INTEGRATION OF MULTI-OMICS DATA WITH GRAPH CONVOLUTIONAL NETWORKS TO IDENTIFY CANCER-ASSOCIATED GENES

Roman Schulte-Sasse

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Berlin, Dezember 2020

Erstgutachterin: **Prof. Dr. Annalisa Marsico**
Zweitgutachter: **Dr. Florian Markowetz**

Tag der Disputation: 30. Juni 2021

SELBSTSTÄNDIGKEITSERKLÄRUNG

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.
Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Berlin, Dezember 2020

Roman Schulte-Sasse

ACKNOWLEDGEMENTS

Many people were directly or indirectly involved in the creation of this thesis and without them it would not have been possible. I foremost want to thank my supervisor, Annalisa Marsico, for her dedication and time during the years of my PhD and for the commitment to the project. Many thanks also to Martin Vingron who has created a wonderful department, full of interesting talks, amazing conferences and great scientific discussions.
I would like to thank Kirsten Kelleher for the organization of the IMPRS graduate school that opened many opportunities and workshops.
I am grateful for the good time I had in the department, especially my office mates Sabrina, Lisa and Stefan. Also, I want to thank Gal Barel and Tobias Zehnder for the friendship, great evenings in Shanghai and for starting to write the thesis together; you made the writing bearable. I am forever thankful to my friends Sarah, Janis, Niklas, Bennet and Ingrid that supported me during the last months that have not been easy. I can't wait to be able to spend time with you again without talking about my work constantly.
I am grateful for my parents who always gave me their support, and Martin Gerhardt without whom I would not have done this PhD.
Finally, I would like to thank Marie for supporting me throughout the thesis and especially the last months. She always encouraged me during this journey and made me laugh in the kitchen even when things weren't going great.

CONTENTS

1 INTRODUCTION

Cancer is not only one disease but rather a set of diseases that are caused by alterations in the genome. Much like evolutionary processes that produce innovations through the pressure of competition, individual cells in the human body are constantly subjected to pressure in their microenvironment. When a cell acquires a growth advantage, it will proliferate faster than its environment, giving rise to a tumor. Untreated, the tumor will grow and interfere with the normal functioning of the tissue it is located in or that of the surrounding tissues, ultimately leading to death. Ever since the discovery that the genome of tumor cells is altered in various ways and especially since the advent of sequencing technologies, studies have tried to discern the multitude of molecular processes that are dis-regulated, hoping to find a cure for cancer.

Unfortunately, a single cure for cancer was not found but instead, large-scale studies involving thousands of patients found an astonishing heterogeneity of tumor cells on the genomic level. The evolutionary mechanisms at play result in remarkably diverse changes that transform cells into cancer cells by giving them capabilities to outgrow neighboring cells, migrate to other tissues and elude programmed cell death. Consequently, a different approach to cure cancer was proposed which focuses on the development of targeted therapeutics that are administered after the tumor was sequenced, termed *precision oncology*. The goal is no longer to find a single cure for cancer but rather a medication for each of its variants. Precision oncology requires a thorough understanding of the processes that lead to the formation of cancer cells in order to develop precise medication and link genomic alterations to an optimal drug combination to prescribe.

The main workforce within a cell are proteins, and genes are the blueprints from which they are made. A central goal in precision oncology is therefore the accurate identification of those genes linked to cancer. With the availability of large and diverse data sets corresponding to various measurements within a cell, machine learning can be used nowadays to predict an association for a gene with cancer malignancies. This can in turn guide hypothesis-driven investigations of promising individual genes towards a more comprehensive set of anti-cancer drugs, a more complete understanding of cancer diseases and finally a decreased mortality of patients.

THESIS OUTLINE

In this thesis, I will present an explainable machine-learning method to predict cancer-associated genes from different molecular readouts of the cell. Chapter 2 will provide a brief background in gene regulation and the various ways by which cells regulate the amount of protein from the roughly $20,000$ genes in the genome and introduce how these regulation programs are disturbed in tumor cells. We will discuss several central ideas and hypotheses from cancer genomics and see that linking a mutation

in the genome directly to a growth advantage of a cell is unsolved as of yet.
Chapter 3 will then present several experimental methods to provide large-scale genomic readouts for a bulk of tumor cells, such as gene expression or DNA methylation. All of the experimental techniques presented here rely on sequencing, the process of extracting the genomic sequence of a sample of cells.
In Chapter 4 we will learn about fundamental concepts and problems of machine learning. At the end of the chapter, graph convolutional networks are introduced, the core of EMOGI, my proposed method for the prediction of cancer-associated genes. One of the key points that distinguish EMOGI from previous approaches conceptually is its capability to integrate very different experimental readouts and the possibility to explain EMOGI's decision-making. Chapter 5 will finally explain how graph convolutional networks are applied to genomic, epigenetic, transcriptomic and proteomic data levels to predict cancer genes. The results of training EMOGI are validated thoroughly on simulated and real-world data in Chapter 6. The power of the proposed method is benchmarked exhaustively against other methods for the prediction of cancer genes and validated on several independent cancer gene sets. The robustness of EMOGI is evaluated in perturbation experiments and the major reasons for the performance of the method are investigated by experiments where individual data types are held out during training. Finally, Chapter 7 will investigate novel predictions made by EMOGI and discern distinct classes of cancer genes that go beyond the classical definitions that describe cancer genes as more often mutated than expected by chance. The results are summarized and put into context in Chapter 8.

2

BIOLOGICAL BACKGROUND

All cells in the human body contain roughly the same genetic material and originate from one stem cell. The information is encoded in a large molecule, the **Deoxyribonucleic acid (DNA)** as a sequence of nucleotides. The ensemble of **DNA** is referred to as the genome and contains a building plan for the cell. In each human cell the genome is split into 23 different chromosomes. The chromosomes reside in the cell nucleus and are typically heavily compacted and very different in size. Specific regions of the genome are genes. Those regions code for proteins which then perform a wide variety of tasks in the cell such as replication, reacting to outside signals and many more. Genes are read by specialized proteins in a process termed transcription. While genes are believed to be the most important part of the genome, they make up only around 2% of it [1].
One of the most fundamental and only partially answered questions in molecular biology is how two cells with the same genome can give rise to completely different cell types with distinct functions and shapes. A neuronal cell, for instance, is very different from a liver cell, suggesting different mechanisms for shaping cell types and fates. Several additional layers of regulation on top of the genetic sequence modify the expression of proteins in the cell in various ways. How the **DNA** is compacted in the cell nucleus differs between cell types, exposing some genes more than others to transcription. Small chemical modifications of the **DNA** sequence allow the gene to be transcribed or prevent transcription. Many more layers regulate cell fate and not all of them have been discovered, but what they have in common is that they establish and maintain tissue-specific levels of proteins in the cells.

Misregulation of these complex processes of genetic and epigenetic regulation can have negative influences on the dynamic and transient equilibrium of protein levels within a cell and ultimately lead to a phenotype or disease. While rare genetic diseases usually have one causal change in the genome (a mutation) that explains the phenotype fully, complex diseases result from combinations of environmental factors, genetic predispositions and mutations acquired throughout the patient's life. Cancer among them is a set of complex diseases, which arise from mutations of the genome but it is usually combinations of mutations that cause the phenotype. They can lead to altered proteins or misregulation of protein levels and ultimately to a cell that grows outside of the strict regulations within the tissue it is residing in. While it was observed that cancer genomes typically divert from the original one, a full understanding of cancer diseases is lacking. This is due to the fact that a change in the genome can produce a phenotype through various ways. When a cell is no longer responding to programmed cell death (apoptosis), for instance, multiple genes or their proteins may be the cause for that and any of the multiple layers of regulation may be responsible. Therefore, linking cancer phenotypes to genotypic changes is hard and requires a deep understanding of regulation of protein expression in the cell.

2.1 DNA, GENE REGULATION & PROTEINS

Most of the dry mass within a cell is protein [1]. Proteins drive almost all processes in the cell and studying how many copies of which proteins are present at a certain moment in the cell is crucial to understanding the molecular basis of these smallest functional building blocks of the human body as well as the emergence of complex diseases. But while only $1-2\%$ of the genome code for proteins [2], much more of it is needed to explain the final protein concentrations within the cell. In the following section, we will see the fundamental principles of gene regulation in order to grasp how small genomic changes can cause mis-regulations in diseases such as cancer.

2.1.1 *The Central Dogma of Molecular Biology*

The **DNA** of living organisms is a sequence of nucleotides consisting of a phosphate, sugar and one of four different bases. The bases are adenine, cytosine, guanine and thymine, abbreviated with their starting letters A, C, G and T. A and T bind to each other as do C and G. Each of the four bases thus has a complement and the **DNA** is present as a double helix in which every nucleotide forms a base pair with their respective counterpart. Hence, the genetic code of one strand of **DNA** can be reconstructed from the other strand. This fact is used during cell division when the two **DNA** strands are isolated and the bases of the second strand are reconstructed one by one from the first one (template strand). The same principle can be used by the cell to detect small changes in the **DNA** by checking if the two strands match everywhere. If not, the cell can try to repair the damage or trigger cell death to prevent the error from having an effect. Roughly 30.000 genes reside within the ~ 3 billion nucleotides of the human genome. Figure 2.1 shows the layout of a typical human gene and while most genes have the elements depicted here, this does not hold true for all genes. A

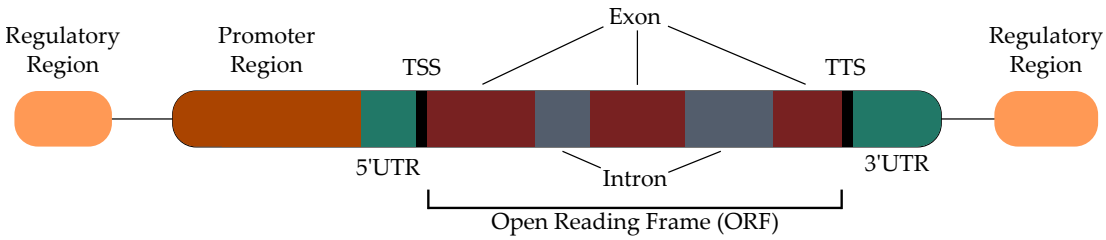


Figure 2.1: **Typical layout of a gene.** The promoter region contains **TF** binding sites and is the place where transcription starts. Genetic alterations here have the potential to change protein levels in a subtle way. The promoter region is followed by an untranslated region (UTR) of variable length. The coding region (ORF) consists of exons and introns. The introns are not translated to protein sequences but exons can be skipped or spliced in different ways, providing yet another layer of regulation that increases the variety of proteins that can be assembled. Regulatory regions like enhancers are located either upstream or downstream of a gene. Assigning a set of regulatory regions to genes is an active area of research and has not been solved yet [3, 4].

gene typically contains a promoter region where the transcription machinery is re-

cruited to initiate transcription. The gene body contains exons and introns which are transcribed into an **Ribonucleic acid (RNA)** molecule. Introns are less important for the final protein sequence because the intronic sequences are removed as part of the post-transcriptional modification process from the newly synthesized **RNA**. Genes are transcribed by the **RNA** polymerase II protein complex (Pol II). This protein complex is recruited to the promoter region of a gene with the help of special proteins called **Transcription Factors (TFs)** and initiates the transcription process. A **TF** is a protein with the capacity to bind **DNA** and thereby influences the rate at which transcription occurs and ultimately the number of **messenger RNA (mRNA)** copies produced in a certain amount of time. Most proteins do not bind **DNA**, however, but rather perform other functions in the cell [5].
The central dogma of molecular biology states that **DNA** serves as a template for **RNA** and the latter serves as a template for proteins [6]. The transcription process produces a single-stranded molecule, the **mRNA**. The **mRNA** is then exported from the nucleus to the ribosome and translated into a protein (see Figure 2.2 for details). A series of three nucleotides (a codon) code for one out of 20 amino acids, the proteins' building blocks. Multiple codons can code for the same **Amino Acid (AA)** such that it is possible for nucleotides to change without changing the **AA** sequence of the protein. The finished protein then either remains in the cytoplasm to act as an enzyme, structural protein and more. However, it can also go back to the nucleus to act as a regulator of transcription (**TF**). Proteins can bind to other proteins, **DNA** or **RNA**, making the formation of feedback-loops possible where proteins regulate the abundance of themselves or other proteins.

2.1.2 *Regulation of Protein Abundance in the Cell*

Cells are highly dynamic microenvironments and proteins are responsible for their correct behavior. Therefore, the number of copies of a specific protein (protein abundance or level) is subject to strict regulation. It is thus not surprising that some proteins are transported back to the cell nucleus after translation to control the amount of **mRNA** produced by their own gene or a set of target genes.
Different cell types have very different desired protein levels and most genes are not transcribed at all in a specific cell type [7].
Many mechanisms of regulating protein abundance occur prior to transcription. The promoter region can influence the expression of the gene through binding of **TFs**. A **TF** binds to the promoter region, often by recognizing a specific sequence, the so-called **Transcription Factor Binding Site (TFBS)**. Thereby it either increases or represses the recruitment of the transcription machinery. The more Pol II is recruited to the promoter region of the gene, the more **mRNA** transcripts are produced and the more protein gets translated.
Although the **DNA** is a linear molecule, it folds in 3D and therefore regions that might be far away from each other in the sequence are actually close in space. Distant regulatory regions, termed enhancers and silencers, can influence the expression of their target genes. They are thought to loop to their target genes and provide binding sites in the **DNA** to which more regulatory proteins, such as **TFs** can bind. Therefore,

TFs binding to the enhancer or silencer can help with the recruitment of the transcription machinery in a similar way as if they were directly bound to the promoter sequence of the gene. Enhancers positively regulate gene expression of their targets while silencers reduce their expression.
Another mechanism for regulating gene expression is the compaction of the genome within the cell. Some regions are very tightly packed, making entire regions of the genome inaccessible. Which regions of the genome are accessible differs significantly between cell types [8].
After the production of **mRNA**, post-transcriptional regulatory mechanisms additionally influence whether the **mRNA** is successfully exported to the ribosome and gets translated there (see Figure 2.2). Small **RNAs**, called micro-RNAs can influence the stability of specific **mRNAs** and either ensure that they are successfully exported or degraded in the cell nucleus [9]. Therefore, post-transcriptional regulation describes yet another layer for the cell to manipulate protein abundances in a fine-grained and highly specific manner.
And finally, small chemical modifications to the **DNA** such as DNA methylation can influence gene expression. DNA methylation describes the addition of a methyl group to the DNA molecule. In mammals DNA methylation occurs almost exclusively at **Cytosine followed by Guanine (CpG)** dinucleotides. In normal cells, the DNA methylation landscape (often called methylome) is highly tissue-specific [10]. While around 75% of **CpG** sites are methylated in mammals [11], important regulatory elements and especially promoters surrounded by **CpG** islands are unmethylated. These differences are usually very sharp and highly similar across cells of the same cell type [10]. High DNA methylation in promoter regions was shown to have a repressive effect on gene expression [12] in two different ways. First, **TFs** cannot bind and transcription is hindered in highly methylated promoter regions [13], and second, high DNA methylation can contribute to the formation of heterochromatin where the **DNA** is so highly condensed that transcription cannot or only rarely occur. The compaction process is caused by specialized proteins (MBD proteins) which then recruit chromatin remodelers to the promoter region of a gene [14].
Unlike the methylation patterns in the promoter regions of genes, gene bodies were associated with highly transcribed genes in the past [10, 14, 15]. The methylome is established and maintained by specialized methyl-transferases, proteins that read, write and remove methyl groups at **CpG** sites in the genome.
The exact concentration of a protein in a cell is highly dynamic and changes quickly over time. It is often determined by multiple layers of regulation. How exactly abundances are maintained and how they change over time is still unclear despite rapid progress in sequencing technologies and new mechanistic insights.

2.1.3 *Proteins Form Pathways*

Proteins are the main building blocks of the cell and execute most of the cell's functions [1]. They are responsible for cell stability (forming membranes, filaments and microtubules), transportation of molecules within the cell, reaction to stimuli from the outside and more. But not surprisingly, proteins rarely work alone and every protein binds to other molecules [1], most often with high specificity. Protein complexes, such

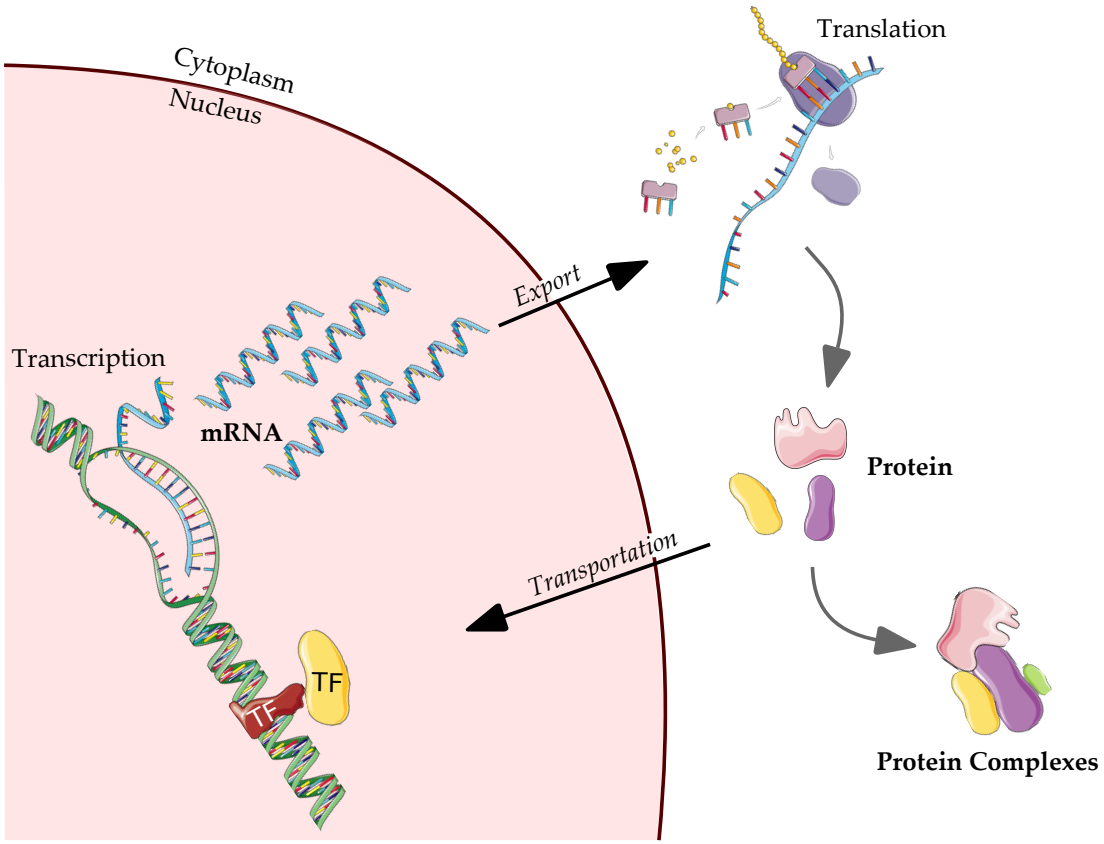


Figure 2.2: **Genesis and life cycle of proteins. DNA** is transcribed into **mRNA** transcripts. Transcripts get exported to the cytoplasm and get translated into proteins. The proteins can either go back into the nucleus to act as regulators of transcription (e.g. **Transcription Factors**) or form protein complexes and perform tasks in the cell.

as the **RNA** Polymerase II transcription machinery mentioned above, consist of more than 10 individual proteins that assemble through non-covalent bonds [16]. Furthermore, long processes like the reaction to outside stimuli require a coordinated action from multiple proteins or complexes. Signal transduction involves receptor proteins that detect the signal and then pass it down to other proteins, causing a cascade at the end of which the cell reacts to the signal by changing transcription rates of genes. Cascades or modules of interacting proteins that pursue a certain goal in the cell, such as signal transduction, cell death or growth, are called pathways. The pathway for programmed cell death, for instance, involves several hundreds of proteins [17] (GO term GO:0097190).

The definition of what constitutes a pathway is vague. They are either defined based on the molecular function that they associate with or from unbiased studies of **Protein-Protein-Interaction (PPI)** networks. In such networks, proteins that can bind to each other are connected, forming an undirected graph that can be studied using mathematical tools (more on **PPI** networks in chapter 3.5).

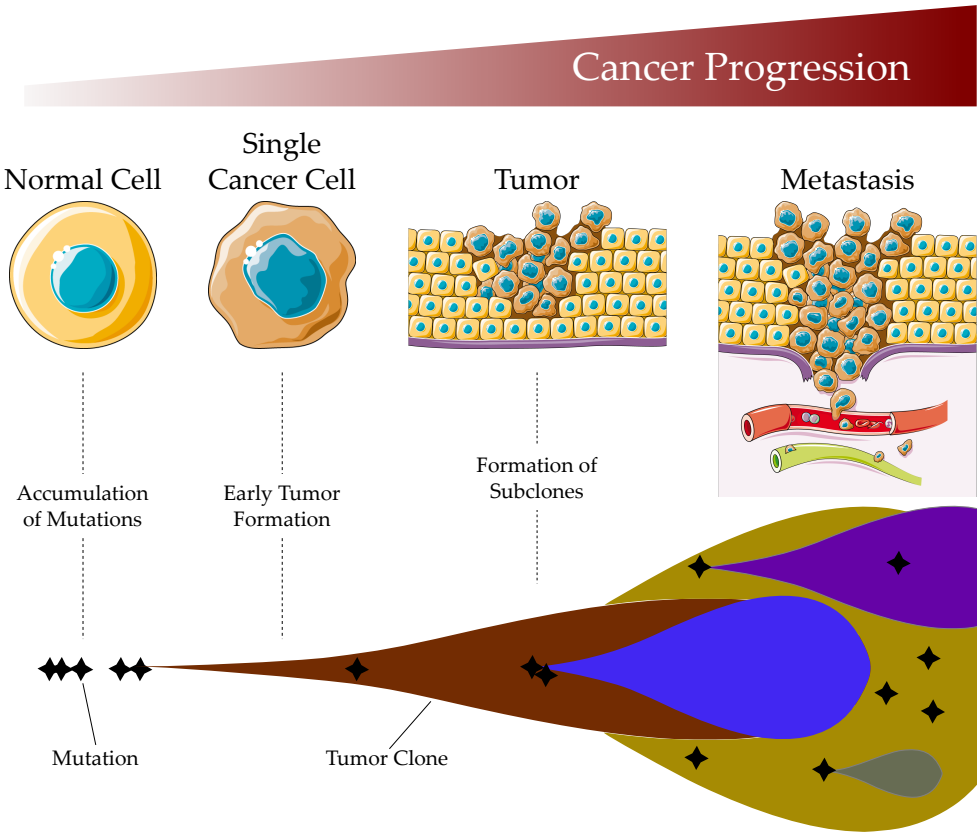


Figure 2.3: **Cancer development without treatment.** Somatic mutations are acquired through errors in cell division or external factors. Driver mutations give abnormal cells a growth advantage, leading to the formation of a tumor. The cancer cells in the tumor will develop further mutations, often at an accelerating rate, leading to the formation of sub-clones with different oncogenic characteristics. At some point, the cancer cells will become able to invade other tissues through the blood stream and metastasize. This is where cancers will become lethal, even if the original tumor was not in a survival-critical tissue.

2.2 CANCER DISEASES

Cancer is the second most frequent cause of death worldwide with a death toll of more than 8 million people per year and describes a set of diseases. It occurs when cells are no longer following their normal programs and start to grow in an uncontrolled manner [18] (see Figure 2.3 for an overview over cancer progression). The accumulation of such "rogue" cells is called a tumor and tumors can start growing in any tissue of the body, giving rise to different tumor types. In their later stages, tumors very often metastasize, invading neighboring and even distal tissues. If not treated, tumors (metastasized or not) will at some point interfere with the normal functions of organs such as the lung, liver or heart and cause death.
Cancer cells that no longer follow their defined programs have altered genomic sequences that lead to abnormal protein sequences or altered concentration of proteins within the cell. Therefore, cancer is a set of diseases of the genome [19]. It is be-

lieved to arise from the accumulation of somatic mutations at random locations in the genome through the lifetime of a patient. A mutation is a small alteration of the genomic sequence (often only one or very few bases) and somatic mutations characterize those mutations not present from birth but acquired and therefore present only in a subset of all the cells in an individual. Another form of more complex genomic rearrangements are **Copy Number Aberrations (CNAs)** which are mutations of larger portions of the genome at once. Different classes of **CNAs** exist. A region can be copied, for instance, leading to a duplication of the sequence within the genome or can be removed from the sequence, leading to a deletion of that region [20–22]. The study of **CNAs** — or structural variation more generally — is an active topic of research and a detailed description is beyond the scope of this thesis. Importantly, **CNAs** can change the number of copies of a gene and influence gene expression when a gene is located within a **CNA**, giving rise to the term.

The reasons why cells acquire mutations are diverse. They can arise during the division of a cell when the exact genomic sequence is not correctly replicated but also from external agents, such as compounds in tobacco, UV light or radiation. Most of the mutations that occur in the genome do not have an immediate effect on the cell. However, once they occur in important regulatory regions or genes, mutations have the potential to change the cell's metabolism in various ways, the simplest being a mutation that changes the **mRNA** product of a gene. This in turn might change the translated protein sequence, potentially giving rise to an altered protein that does not function correctly anymore.

As depicted in Figure 2.3, the accumulation of mutations in a cell continues. The process often accelerates because crucial cell functions such as **DNA** repair, replication and programmed cell death are no longer functioning correctly until the cell ultimately transforms into a cancer cell. Cancer cells are cells which grow outside of the tightly regulated microenvironment of the surrounding tissue. In order to do so, cancer cells must change the normal metabolism of the cell to no longer react to outside signals such as programmed cell death (apoptosis), divide more often than other cells and acquire more energy to divide very often [23]. It is still unclear how exactly the transformation process happens. Also, estimates on how many alterations are required for a cell to become cancerous vary greatly [22, 24–26]. Molecular profiling has revealed that most somatic mutations in cancer genomes occur outside of genes [27]. But due to the complex regulatory mechanisms, non-coding mutations can still influence the expression of genes and hence aid the transformation of normal cells to cancerous ones (tumorigenesis) [27, 28].

Despite the mutation generation process being random, the genomic alterations in cancer cells carry much information about important elements for tumor progression and initiation [25]. Cancer cells need a growth advantage over other cells in order to form a tumor and metastasize. Genes that are more often mutated than one would expect are probably associated with cancer or cancer-related cell functions [25, 29] because the mutations would not be present so often if they did not confer a growth advantage. Large-scale studies of cancer genomes such as those from **The Cancer Genome Atlas (TCGA)** [30] have not only found mutations that occur in many patients but also several genes to be more often mutated than expected, giving rise to the notion of cancer driver genes. Cancer driver genes are often defined as genes that,

when mutated, increase the net cell growth *in vivo* [31]. Due to the complex nature of gene regulation in humans, however, a gene is not required to carry a mutation to be implicated in cancer diseases.

2.2.1 *Personalized Medicine*

The stochastic way in which mutations accumulate in a genome makes treatment of cancers very hard. While some drugs specifically target hallmarks of cancer [23], others specifically target proteins to reestablish normal cell functions. The later a tumor is detected, the more diverse its mutational landscape has become [26, 32] because once the cell has lost its genomic stability, mutations and genomic rearrangements occur at a much higher rate. The different sub-populations within a tumor each have their own vulnerabilities and resistance, hindering treatment [33].
The goal of personalized medicine is to sequence tumors and then decide on a treatment based on the mutational composition of the tumor. Therefore, a deep understanding of cancer diseases as well as a plethora of very specialized drugs is needed to cure patients. Examples of specialized treatments target *HER2* overexpression in breast cancers through kinase inhibitors and specialized antibodies [34] or target the *BCR-ABL* fusion gene in Chronic Myeloid Leukemias through tyrosine kinase inhibitors [35]. By now, personalized treatments for many cancer types are available [36]. Most of the personalized cancer treatments target proteins that are either overexpressed, mutated or that are important for the growth of cancer cells but not required in normal cells [37]. To develop a highly specialized battery of drugs, it is therefore crucial to completely catalogue all genes with an association to cancer. Only then, specialized treatment can be administered to restore normal cell functions, kill all cancerous cells of a tumor and prevent relapse.

2.2.2 *Finding Cancer Genes*

Multiple studies have found cancer drivers in a hypothesis-driven way [38, 39] where a gene was believed to have a specific function that relates it to cancer and that association was tested in a laboratory. With the advent of mutation profiling data for thousands of patients by **TCGA** or—more recently—the **International Cancer Genome Consortium (ICGC) Pan-Cancer Analysis of Whole Genomes (PCAWG)** [28], computational methods have emerged to statistically predict cancer driver genes from molecular data sets [31, 40–44]. Early computational and experimental studies for detecting cancer driver genes have focused on genes that are mutated more often than expected by chance, leading to the discovery of *EGFR*, *KRAS* or *MYC* genes [19].

However, as seen in Section 2.1.2, multiple layers of regulation can cause changes in protein concentrations. Hence, over-mutated genes only explain a fraction of the underlying causes of cancer [19, 25, 37]. While a few genes are highly mutated, there often exists a "long tail" of the mutation rates that contains less frequently mutated genes which nonetheless influence cancer progression [19]. This partly comes from the organization of proteins in pathways and complexes (see Section 2.1.3 for details) where the normal cell functions can be modified or disabled by mutating any of the

genes in the pathway. As a consequence, each of the genes within the pathway is only mutated in a fraction of patients but the cellular function is disabled in all or many of them. Different strategies have been adopted to account for that heterogeneity. One way is not to consider different cancer types individually but rather in a joint analysis (termed pan-cancer analysis) [30]. If somatic alterations are present in multiple cancer types, their statistical significance is increased, allowing to distinguish passenger and driver mutations even for rarely altered genes. The rationale behind pan-cancer analyses is that there are common alterations across cancer types and that rationale is supported by evidence in many cases. It was, for instance, shown in multiple studies that tumor cells often exhibit stem cell-like behavior and reactivate the same developmental genes across different cancer types in order to outgrow neighboring cells [45–48], justifying pan-cancer approaches especially for rare or understudied tumor types.

Another orthogonal strategy that computational methods have started to adopt is the incorporation of **Protein-Protein-Interaction** data into the algorithms [41, 44, 49]. This way, alterations in a protein complex or pathway can be aggregated and significantly mutated modules were shown to correspond to known cancer pathways [41, 42, 50].

Both of the above-mentioned causes for heterogeneity only partly explain the genomic landscapes of tumor cells, unfortunately [25, 51]. Regulatory regions such as enhancers can regulate gene expression via **TFs** and can equally be disrupted in cancer [28, 37, 52]. If binding sites of **TFs** are disrupted by a mutation, this can have an effect on the expression of the target gene(s) of that regulatory region. As of yet, the exact locations of regulatory regions determining the expression of a gene of interest are not entirely known, making it hard to assess the exact consequences of mutations occurring outside of genes despite recent efforts [52, 53]. On top of that, genomic data of whole genomes of cancer patients only recently became available on a large scale with initiatives such as **PCAWG** [28]. Therefore, to accurately understand the impact of non-coding mutations in cancer, one has to decipher the complete set of regulatory mechanisms because mutations can affect every layer of genome regulation.

Fortunately, it is possible to measure the abundance of **mRNA** in a cell experimentally, making it possible to find putative cancer genes that are not mutated but nonetheless show altered expression in tumor samples. Unlike the genome which is the same across all cells (not counting somatic mutations), each cell type expresses different genes at different levels in a highly dynamic fashion, making it harder to quantify expression changes between tumor and normal samples [3, 54, 55]. Furthermore, **mRNA** expression is only an intermediate measure of the protein expression, not accounting for post-transcriptional regulation through micro-RNAs (miRNAs) or other mechanisms. Nonetheless, some of the variability in protein expression through non-coding mutations and epigenetic aberrations can be explained through **mRNA** expression and multiple studies have found cancer genes that are overexpressed but rarely mutated [56–59]. One of the prototypical examples of such a gene is the transcription factor family of *MYC* genes which are rarely mutated but often amplified (targeted by a copy number duplication event) and overexpressed. Due to their role as **TFs**, *MYC* overexpression leads to increased expression of target genes, many of which play important roles in cell proliferation [20, 37, 60]. As was observed already

for somatic genomic lesions, several genes were observed to be recurrently expressed differently in different cancer types [37]. Following the rationale from above, a pan-cancer approach therefore is expected to help identify genes that are only differently expressed in a small part of the cohort, and that would be missed by investigating single cancer types.
Lastly, alterations of the genome are not the only changes in cancer cells. Epigenetic aberrations can influence gene expression and provide yet another mechanism of how cancer cells corrupt normal cell functions. DNA methylation, introduced in the previous section, is significantly altered in cancer genomes [61]. Mutations in readers or writers of methyl groups at **CpG** sites (or mutations in their regulatory regions) cause changes in the methylome. It has been observed in multiple cancers that the normally sharp patterns of DNA methylation are dissolved. This phenomenon of global hypo-methylation is accompanied by a local hyper-methylation of selected **CpG** islands [61, 62] and silences multiple genes. The broad changes of the methylome in cancers most often lead to the deactivation or activation of cancer driver genes [10, 12, 62, 63] and hence, measurements of DNA methylation are important features for the identification of cancer driver genes. In colorectal cancers, for instance, up to 800 genes are transcriptionally silenced through aberrant DNA methylation compared to normal adjacent tissues [64–66]. As for genomic lesions and differential expression, many cancer-associated genes were reported to show differential methylation patterns across multiple cancer types and especially that cancer cells show stem cell-like behavior with certain developmental genes frequently being reactivated in cancer cells [45, 47, 48, 67]. While the DNA methylation landscape is highly tissue-specific, differentially methylated promoter regions are enriched for developmental genes across multiple cancers [46]. Therefore, a pan-cancer approach is likely to give a stronger signal of differentially methylated promoter regions compared to analyses of individual cancer types and expected to better recapitulate the activation and/or maintenance of the core pluripotency network through aberrant DNA methylation.

2.3 COMPUTATIONAL METHODS TO PREDICT CANCER GENES

To date, a plethora of computational tools has emerged to predict cancer-associated genes but often adopt very different definitions of cancer genes. MutSigCV [40] and 20/20+ [31] are representatives of methods that predict cancer genes based on somatic mutations accumulated within the gene body. They follow the rationale that genes which harbor frequent driver mutations confer a growth advantage to the cell and the main focus of the methods is to distinguish driver mutations from random passenger mutations. MutSigCV, for instance, constructs a sophisticated background model that accounts for gene length and nucleotide composition for the prediction of driver genes. 20/20+ is then even capable of differing between **Tumor Suppressor Genes (TSGs)** and oncogenes from the distribution of mutations within a gene.
More recently, computational methods have started to model the heterogeneity caused by pathways and other protein interactions through the use of **PPI** data. HotNet, HotNet2 and hierarchical HotNet [41, 42, 68] attempt to find modules of frequently mutated genes within **PPI** networks and found them to correspond to important cancer pathways. Such methods often aim to identify entire modules instead of single genes,

following the idea that the exact gene that is altered may vary from patient to patient but the cellular function does not [69]. Recent advances further account for node degree biases in the HotNet algorithms through specialized normalization [70].
All of the above-mentioned methods only use mutation data for their predictions. Recently, predictive methods have started to incorporate different *omics* levels in their predictions, often using **Machine Learning (ML)** for that [44, 49]. ModulOmics [49], for instance, solves a linear integer programming problem to predict modules of cancer genes from genetic, transcriptomic and protein interaction data and regulatory information about **TFs**. And LOTUS [44] integrates mutations alongside **PPI** information and leverages a supervised **ML** approach to predict oncogenes and **TSGs** separately.
Finally, several methods have been developed to solve similar problems, such as grouping samples based on molecular profiles to identify subtypes of cancers or predict survival and severity of patients [71]. While these problems are also highly relevant, they are not in the scope of this thesis.

2.4 SUMMARY

Cancer is a set of highly complex diseases originating from genomic alterations (mutations, structural variation and even chromosome duplications). Tracing all genomic changes to a phenotype is theoretically possible but requires a rather complete understanding of gene regulation. The final level of a protein within a cell depends on many factors and a complex regulatory landscape that is not completely understood as of yet. Cancer cells typically "hijack" many different regulatory mechanisms in an evolutionary manner to gain growth advantages over neighboring cells. Mutations in the readers or writers of epigenetic marks, for instance, can cause genome-wide alterations, activate oncogenes or inactivate tumor suppressor genes [62]. Non-coding mutations can suppress or activate genes when occurring in regulatory regions of those genes [27], and mutations in **TFs** can alter the expression of many target genes simultaneously [37]. However, multiple molecular readouts such as epigenetic modifications and **mRNA** expression can be used to associate cancer phenotypes with genes when the regulatory processes underlying the phenotype are not fully understood.

3 EXPERIMENTAL TECHNIQUES

We are drowning in information and starving for knowledge.

— John Naisbitt [72]

We have now seen that it is hard to decipher the regulatory mechanisms and dynamics that control the highly different levels of protein abundances in all cells of a human body with the same underlying genome. This chapter will briefly introduce experimental methods to not only sequence the genome but also directly measure intermediate products that help find mis-regulated cell states without having to fully understand the regulatory landscapes that give rise to different cell types and protein levels.

3.1 HIGH-THROUGHPUT SEQUENCING

The **DNA** sequence is the most fundamental molecular readout because it is the same across tissues and is not changing over time except for somatic mutations. Many approaches have been developed to sequence the genome or parts of it, starting with Sanger sequencing [73], evolving to Microarrays [74] and now **High Throughput Sequencing (HTS)** (see [75] for a review of the history of **DNA** sequencing methods). Today, **HTS** (or next-generation sequencing, NGS) is the most prominent way to sequence **DNA** in a sample (a bulk of cells). Different **HTS** sequencing methods exist with Illumina dye sequencing being the most popular among them. An additional overview of currently used next-generation sequencing approaches with their advantages and disadvantages is given in Goodwin et al. [76].
In brief, most **HTS** methods work by first fragmenting the **DNA** molecule to obtain many short nucleotide sequences, followed by a heavy amplification of the same [77]. Once amplified, sequencing uses only a single strand of the short **DNA** sequences and reconstructs the second strand using polymerase as in normal cells. This enzyme reconstructs the second strand of the **DNA** sequence that is to be read, using specialized nucleotides with fluorescent markers. Once one of such nucleotides is attached to the newly synthesizing **DNA** strand, a camera takes a picture and uses wavelengths and intensity to determine which nucleotide was just sequenced. High throughput is achieved by performing the sequencing reactions massively in parallel. To identify nucleotides with high accuracy, the same piece of **DNA** is heavily amplified and then sequenced thousands of times [2]. The sequencing can be targeted to regions of interest, such as exomes [78] (termed **Whole Exome Sequencing (WES)**) or specific genes.
The results from **HTS** are short reads (typically $\sim$ 150 nucleotides long) that have to be mapped computationally to a reference genome (see Figure 3.1). While the reads are relatively short, they are highly accurate (correct in $\sim$ 99.9% of the cases), making the detection of point mutations (one changed nucleotide) very reliable.

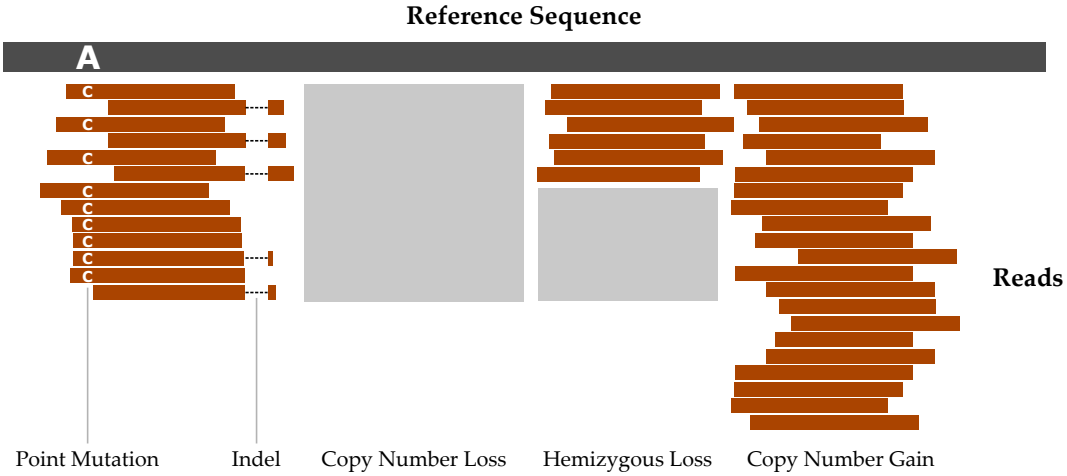


Figure 3.1: **Types of genomic alterations/rearrangements detectable through HTS.** Different types of genomic changes can be detected from **HTS**. Point mutations show multiple reads with nucleotides being different compared to the reference genome. Small insertions and deletions (indels) can be detected when multiple reads leave out a part of the reference or have extra nucleotides added. Larger chromosomal rearrangements, such as copy number changes can also be detected. Figure adapted from [79].

3.2 MUTATION PROFILING

High throughput methods can be used to detect genomic alterations in tumor cells and have become the state of the art in cancer genomics studies and clinical applications [28, 79]. **Single Nucleotide Variants (SNVs)** are derived from sequencing tumor and normal samples in parallel to exclude germline mutations and other divergences from the reference genome. The capacity to detect **SNVs** is highly dependent on the sequencing depth, that is how many bases are sequenced compared to the number of bases in the starting material ($\sim$ 3.2 billions for whole human genomes, $\sim$ 30 millions for human exomes). Mutations are then called by computational tools that count the number of reads showing the alteration and compare it to sophisticated background models [80, 81] (see Figure 3.1). The challenges in calling mutations in cancer genomes are that usually samples are not pure (immune cells, blood vessels and other cells are mixed with tumor cells) and that different subpopulations in a tumor can be mutationally very heterogeneous [19, 82]. This leads to mutations occurring at low frequencies despite their importance in tumor formation, requiring greater sequencing depth to distinguish them from experimental noise.

Next-Generation Sequencing (NGS) can also be used to detect structural changes in cancer genomes, such as copy number changes and small indels [83] (see Figure 3.1 for an overview of genomic changes detectable from **NGS** read data). Again, the challenge in reliably detecting copy number aberrations in cancer genomes is a statistical one and is usually tackled with computational methods that decide whether the reads provide enough information about structural changes or not [20, 83, 84]. Paired-end reads help with the ability to detect structural variation in cancer genomes [82]. For

a review of the applications of **NGS** methods to detect mutations and chromosomal rearrangements, see Meyerson et al. [79].

3.3 METHYLATION PROFILING

DNA methylation describes the addition of methyl groups to the **DNA** molecule. In humans, DNA methylation occurs almost exclusively at **CpG** dinucleotides. DNA methylation represents the most extensively studied epigenetic modification and has been defined as a major hallmark of cancer [85]. As discussed in Section 2.1.2, the DNA methylation landscape (the methylome) is controlled by proteins that read and write methyl groups to the **DNA**. Somatic mutations in their genes (or regulatory regions of them) can introduce aberrant methylomes in cancer cells.
The most common approach to measure DNA methylation is bisulfite treatment and conversion. Upon treatment with bisulfite, unmethylated cytosines (C) are changed to uracil during library preparation. Uracil (abbreviated U) is normally only present in **RNA** molecules and is read as a thymine (T) during sequencing. After the bisulfite treatment, **NGS** is performed and the read counts are used to compute the fraction of cells in the sample in which the cytosine was methylated. The resulting average is called β-value and ranges from 0% (the **CpG** site is methylated in none of the cells) to 100% (the **CpG** site is always methylated). There are different experimental protocols for measuring DNA methylation with specific advantages and drawbacks which are reviewed in [86]. The most widely used platform that allows for a larger cohort is the Illumina Infinium 450k Methylation array based on microarrays. It sequences $> 450,000$ targeted **CpG** sites in the genome that were chosen to either lie in promoter regions, gene bodies, regulatory regions or additional regions that have been reported to be differentially methylated in the past [87, 88]. Each experiment with that platform will give β-values for the exact same bases in the genome, allowing comparison between different samples. However, that approach only covers 1% of **CpG** sites in the genome [86].

3.4 GENE EXPRESSION QUANTIFICATION

Another shortcut to finding mis-regulated genes and proteins in cancer — thereby bypassing the complex mechanisms of gene regulation — is a readout of the numbers of copies of a gene product (**mRNA** or protein). While the number of copies of a protein in a cell can be directly measured as well [89], this process is harder and usually produces data for only some proteins [89]. However, the **mRNA** levels can be measured through **NGS** methods in a high-throughput manner, making this approach widely used. The **mRNA** levels then serve as proximal measurements of protein expression, despite not acknowledging post-transcriptional regulation through various mechanisms.
The number of copies of **mRNA** transcripts in a cell are measured using **RNA sequencing (RNA-seq)** [90]. This experimental protocol first extracts **RNA** in a cell and fragments it. Then, the **RNA** is reverse-transcribed to **complementary DNA (cDNA)**, adaptors are added and the fragments are amplified and subsequently sequenced

with **NGS** (typically Illumina protocols but other approaches exist [91]). The resulting reads are mapped to a reference genome (if available) and the read counts for a transcript serve as a measure of **mRNA** abundance. Reads are typically further processed to account for artifacts occurring during amplification or adapter ligation and final read counts are often expressed as **Fragments Per Kilobase Million (FPKM)** to normalize for read length and sequencing depth. For an overview of different protocols for **RNA-seq**, see the recent review by Stark et al. [54].

3.5 PROTEIN-PROTEIN INTERACTIONS

In Section 2.1.3, we saw that proteins interact with other proteins or molecules such as **DNA** or **RNA** and even organize in pathways to perform their function in the cell. These interactions make it harder to find putative cancer genes because changes in many proteins of a complex or pathway can cause the phenotype, no matter which protein is affected by the change. Measuring how proteins interact in a cell gives an approximation of pathways and complexes that are related to a specific protein of interest. Interactions between proteins are believed to be a transient and dynamic process that changes with different needs of the cell and any experiment finding two proteins interacting reports a snapshot of such a process. Therefore, experimentally measured **PPIs** describe the possibility of two proteins interacting rather than an interaction taking place in all cell types and at all times.

Experimental methods to find **PPIs** differ in two approaches. Binary methods like **Yeast Two-Hybrid (Y2H)** directly measure the interaction partners of a protein of interest [92]. In that system, the first protein (bait) is fused to the DNA binding domain of the yeast **TF** *Gal4* while a second protein (prey) is fused to the activation domain of the same **TF**. When the reporter gene of *Gal4* is subsequently expressed, the two proteins interact. Conducting many of those experiments in parallel allows studying all or most interaction partners of the protein of interest. In **Y2H**, only direct interactions between two proteins of interest are measured and false positives are typically low. However, the interactions are measured in a yeast host, thereby drastically changing the environment.

Co-complex methods, on the other hand, like **Tandem Purification coupled to Mass Spectrometry (TAP-MS)** or co-immunoprecipitation, attempt to measure all the interactions between the bait protein and its interaction partners at once. In **TAP-MS**, a tagged protein is isolated along with all of its interaction partners from a solution by immunoprecipitation and the resulting protein mix is then divided into smaller peptides and subjected to mass spectrometry analysis. Co-complex methods also measure indirect interactions that originate because protein A interacts with protein B and B with C and will then report a false interaction between A and C [93].

Nowadays, databases hold **PPIs** for a large variety of proteins such that an entire network for an organism can be constructed. Such **PPI** networks usually contain interactions from different platforms and experiments and meta-databases unify several resources to produce high-confidence interaction networks [94, 95].

4 COMPUTATIONAL BACKGROUND

All models are wrong but some models are useful.

— George E. P. Box

In the previous chapters, we have seen that linking genotypic changes to cancer phenotypes is a great challenge. At the same time, intermediate measures of gene products or epigenetic marks can be obtained with modern sequencing technologies. In order to detect cancer-associated genes without a complete understanding of genetic regulation, we can use a computational approach that integrates different heterogeneous data sets to predict whether a gene is or is not associated with cancer phenotypes.
In this chapter we will learn the computational prerequisites for EMOGI, an explainable multi-omics graph integration approach that uses such heterogeneous data types to predict cancer-associated genes.

4.1 LEARNING FROM DATA

ML describes a field of study operating at the intersection of computer science, statistics, and engineering. Methods and algorithms assigned to the field of **ML** are able to learn complex patterns from a given data set. "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." [96]
The task T is highly problem-dependent and can vary a lot, depending on the application. A valid task could be the recognition of human faces in images [97], directing the motors of a robot such that the robot walks without falling down [98] or the detection of cancer genes from molecular data sets. The experience E in modern data-driven machine learning is usually represented by data points (or examples) in which a data set X consists of individual data points x_i. Data points are mostly represented as vectors $x_i \in \mathbb{R}^p$ where p denotes the number of features per data point. The terms data points and feature vectors will be used interchangeably to denote x_i throughout this thesis. Mostly, learning algorithms have the entire data set X at their disposition and designers of such algorithms can freely choose to learn from one example at a time or from all at once.
The performance measure P describes a metric that defines how well the algorithm is currently achieving its goal and when it should stop learning. The performance measure is usually defined by the learning problem [99]. In the case of a robot that is learning how to walk, performance could be described by the distance without falling. In case of image recognition, on the other hand, performance might be accessed through expert-curated data where the algorithm's prediction is compared to a label that was assigned by an expert to each image (a process called labeling). Most often, a performance measure P can be derived more easily when many labeled data

points are available. In computational biology, however, the performance measure is often not an obvious choice because the goal is to generate knowledge which is only useful when the domain of the application is not very well understood. In addition, labels are often hard to come by in the field [100]. In some cases, experimental validation on a large scale can provide labels [101–103] but such studies are not always available and often, the validations have to be done under slightly different conditions than the application (use of cell lines instead of human tissues for knockdown experiments, for instance).

Different learning algorithms have been developed for various types of applications.

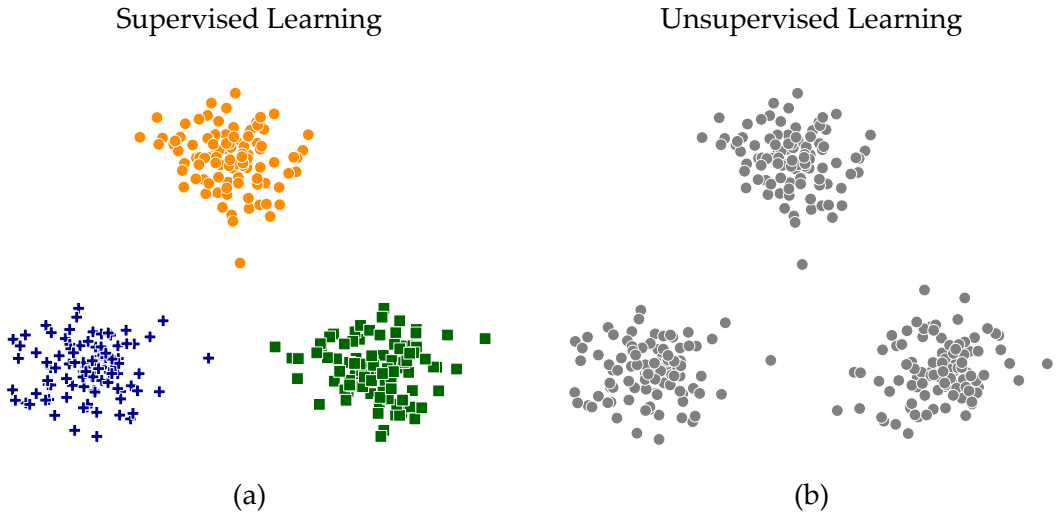


Figure 4.1: **Supervised & unsupervised learning. a** In supervised learning, the learning algorithm is provided not only with data points x_i but also with labels y_i that make the learning easier. The labels take three different labels for three distinct classes in this example. **b** For unsupervised learning, only the data points x_i are given and the algorithm has to distinguish the classes without any labels. In the example, a distance metric (like the Euclidean distance) can be used to distinguish between points that are close to each other and more distal points (the process is known as clustering).

They can mostly be distinguished into supervised and unsupervised methods, depicted in Figure 4.1. Supervised methods are the focus of this thesis, although unsupervised machine learning methods are used in Section 7.2. Supervised **ML** algorithms use, in addition to the data points x_i, a provided label (or target) y_i, which carries information on the true prediction outcome of a data point. This can be a class label (an image contains a human face or not), a continuous value (the predicted expression level of a gene) or even a vector of values. Hence, supervised learning algorithms learn a mapping from x to y, using the data set $D = \{(x_i, y_i\}_{i=1}^N$. In supervised learning tasks, the performance measure can easily be defined as the divergence of the algorithms prediction $\hat{y}_i$ from the true label y_i.

Unsupervised methods, on the other hand, lack the label and therefore have to find structure in the data itself. An example of unsupervised learning is clustering, where the goal is to find groups of similar data points in an automated fashion, for instance correctly inferring three groups of data points in Figure 4.1 b. Unsupervised learning

is thought to be inherently harder and often require more assumptions about the underlying data generation process [104].

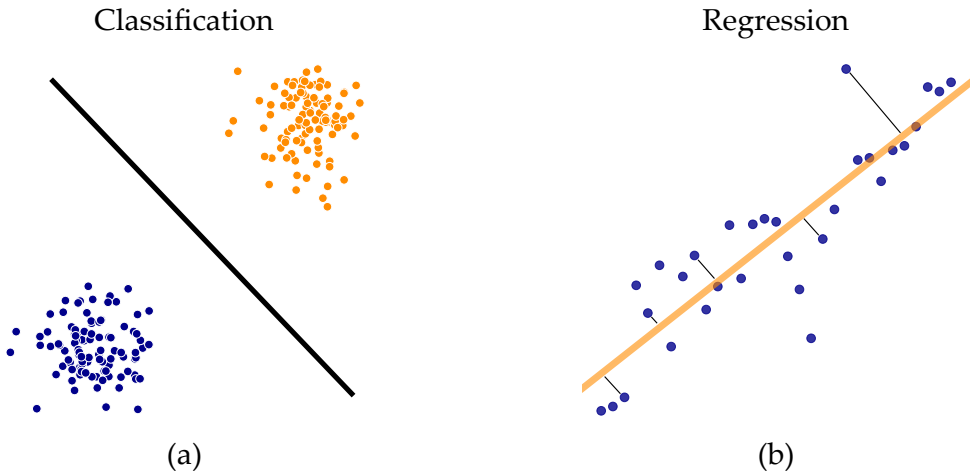


Figure 4.2: **Classification and regression.** The different problems of classification and regression are depicted in two panels. **a** In classification, the goal is to distinguish classes of points. In a binary classification setting, a hyperplane (a line in 2D and generally a $p-1$-dimensional plane in a p-dimensional problem) serves as the decision boundary between the two classes (depicted as black line). **b** In regression, the goal is to find a function that minimizes the distances of blue points to the line (distances are depicted as black lines for some points). Here, the data is assumed to lie on a line with some added error (the problem is said to be linear).

Supervised learning problems can be further divided into classification and regression problems. In classification, the target value y_i denotes a class label and there is a discrete number of classes. The simplest setting, called binary classification, distinguishes between two classes of data points. The concept of regression, on the other hand, describes the problem of predicting continuous values. Predicting the future value of a stock or the expression level of a gene are examples of regression problems. Figure 4.2 depicts the conceptual difference between classification and regression. While the former requires finding a decision boundary that separates the classes, the latter has to fit a function through the data points in order to make predictions for new data.

4.1.1 *Linear & Logistic Regression*

The two concepts of classification and regression are closely related in the setting of linear regression and logistic regression. Linear regression describes an algorithm for fitting a line through data points such that the line is most probably the one that generated the data. The optimal solution for a one-dimensional case is depicted in Figure 4.2b. The x-axis shows the feature x_1 of the problem, while the y-axis shows the output y. To find a linear mapping between feature and output, linear regression attempts to find a line that minimizes the distance of each data point to the line. The assumption behind linear regression is that the data was generated by a linear

process where $y = ax + b$ with some error e. The error e is denoted by the black lines in Figure 4.2b and minimizing e corresponds to an optimal line that models the data generation process.
Logistic regression, on the other hand, describes a classification method, despite its name. It uses the general idea of a linear regression and embeds it into a binary classification scenario. Here, the response variable y is a class variable, encoding the membership of data points to either class 0 (orange) or class 1 (blue).

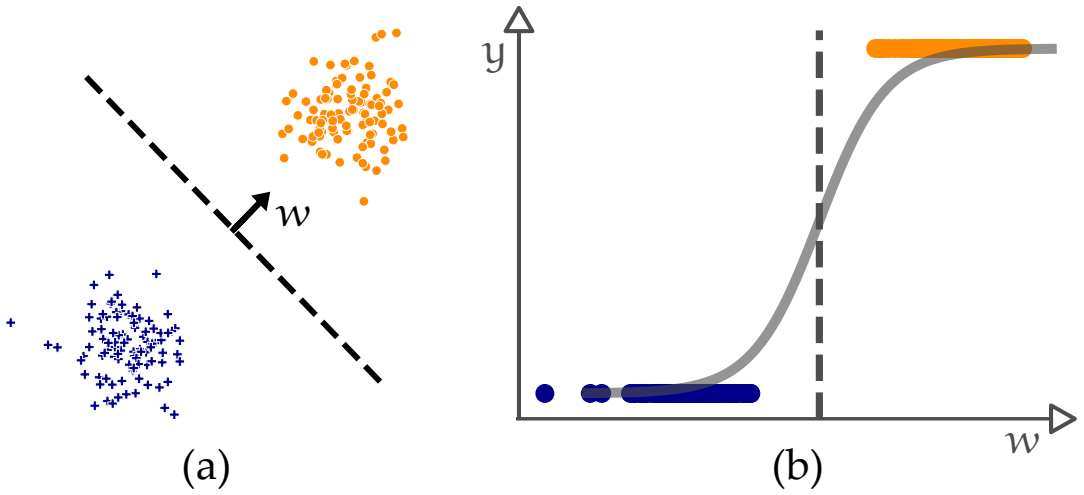


Figure 4.3: **Logistic regression classifies by projecting the data on a weight vector w and then applying a logistic function. a** The binary classification problem from Figure 4.2b is depicted again. The weight vector w is learned during the training process. All data points are projected onto w. **b** The data points projected on w are depicted on the x-axis while the class labels y (0 and 1, encoded by blue and orange points) are depicted on the y-axis. A logistic function is applied to the projected points to assign probabilities to the data points. Dashed lines indicate the decision boundary between the classes.

Fitting a linear regression through the data points similar to Figure 4.2a, however, is not ideal because of a linear increase of certainty that the point belongs to class 1 (the y-axis in Figure 4.3a) and no meaningful range of the output data. Ideally, a classification method should output a probability that a point belongs to class 1 (the orange points) and this probability should not increase linearly because the region around the orange points should be assigned a probability close to 1. Only the region close to the decision boundary should be ambiguous. Therefore, logistic regression replaces the linear function $f(x) = ax + b$ with the logistic function $\phi(x) = \frac{e^x}{1+e^x}$, giving the logistic regression its name. The successful fitting of a logistic function onto the one-dimensional data from before is depicted in Figure 4.3b. It assigns high probabilities to the orange points and low probabilities to the blue ones. In between there is a steep decrease in probability, such that the assigned probabilities around the decision boundary vary a lot but stay stable in the well-defined regions.
Training a logistic regression model — and in fact, training most linear models — corresponds to finding a weight vector w that projects the data to a lower-dimensional subspace (the two-dimensional problem from Figure 4.3a is projected on a one-dimensional

weight vector in Figure 4.3b). The projection of x onto w makes it possible to find a simple decision boundary (dashed lines in Figure 4.3) that indicates a switch between the two classes.
Unfortunately, there exists no closed-form solution to find the optimal parameters of a logistic regression problem. Therefore, optimization is mostly done through maximum-likelihood estimation. This procedure tries to find the parameters w of a model such that the probability of observing the real data is maximized. For a logistic regression problem, the maximum-likelihood estimate can be written as:

$$\mathcal{L}(w) = \prod_{i=1}^{N_1} \Pr(x_i, w) \prod_{i=1}^{N_2} \big(1 - \Pr(x_i, w)\big) \tag{4.1}$$

where $\Pr(x_i, w) = \phi(w^T x) = \frac{e^{w^T x}}{1+e^{w^T x}}$ and N_1 denotes the number of data points for class 0 while N_2 denotes the number of data points for class 1. When we now compute the log-likelihood (products are transformed to summations in the logarithm, making calculations easier and alleviating numerical issues), we obtain:

$$\begin{aligned}\log(\mathcal{L}) &= \sum_{i=1}^{N_1} w^T x_i - \sum_{i=1}^{N_1} \log\left(1 + e^{w^T x_i}\right) - \sum_{i=1}^{N_2} \log\left(1 + e^{w^T x_i}\right) \\ &= \sum_{i=1}^{N} y_i w^T x_i - \sum_{i=1}^{N} \log\left(1 + e^{w^T x_i}\right)\end{aligned}$$

In theory, setting the derivative of $\mathcal{L}$ to 0 will denote extreme points in the maximum-likelihood estimation that denote optimal weights w. Due to the non-linear function and the exponential, however, it is hard to obtain an analytical solution. Therefore, gradient ascent is typically used to compute optimal weights iteratively by moving in the direction of the gradient.

4.1.2 *Gradient Ascent & Gradient Descent*

Gradient descent and ascent are popular iterative optimization algorithms. Gradient descent finds extreme points of a function when the function is partly unknown but differentiable.
Gradient descent operates by starting with an initial random guess of the parameters w and computing the gradient of the function at that initial point. It then updates w by taking a step into the direction of the steepest descent or ascent as indicated by the gradient. This procedure is repeated until some criterion of convergence is met.
For the logistic regression, the gradient can be computed as:

$$\nabla\mathcal{L} = \sum_{i=1}^{N} \left(y_i x_i - \frac{e^{w^T x_i} x_i}{1 + e^{w^T x_i}}\right) \tag{4.2}$$

$$= \sum_{i=1}^{N} \left(y_i x_i - \Pr(x_i, w) x_i\right) \tag{4.3}$$

$$= \sum_{i=1}^{N} x_i \big(y_i - \Pr(x_i, w)\big) \tag{4.4}$$

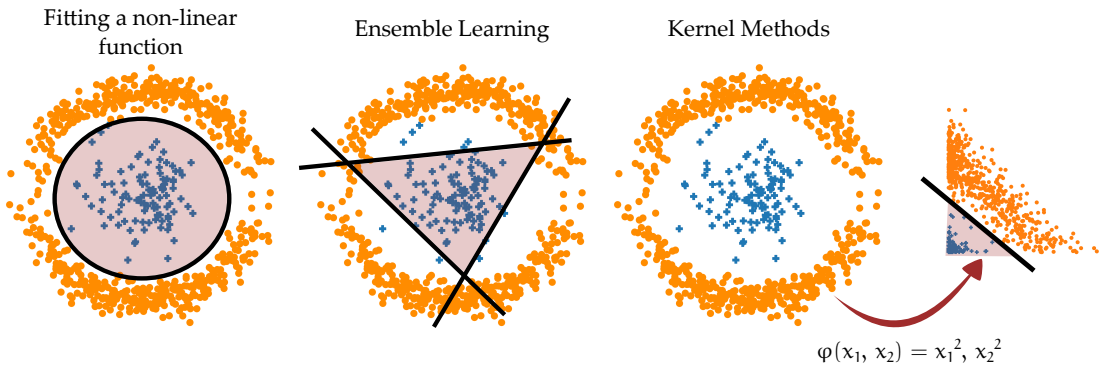


Figure 4.4: **Different approaches for non-linear problems.** The most straightforward solution to deal with non-linear classification problems is to explicitly model a non-linear function. Ensemble methods fit several weak classifiers (typically linear ones) that each solve a part of the problem. Together, the weak classifiers can model the problem as a whole. Finally, data transformations make the data linear prior to using a classification algorithm. Most often, data points are mapped to very high-dimensional spaces to ensure linear separability.

The gradient descent algorithm now proceeds by updating w according to:

$$w = w - \eta \nabla \mathcal{L} \tag{4.5}$$

until convergence. Gradient ascent replaces the minus sign with a plus sign to move into the direction of the steepest ascent. This process is not guaranteed to yield an optimal solution for arbitrary functions because the optimization can get caught in local extrema. Many different variants of the gradient descent algorithm have been developed, summarized in [105] with some variants converging faster or being able to overcome small local extrema. For some functions, such as the loss function from logistic regression, however, the results from gradient ascent are optimal and unique [106] but even when this is not the case, results from gradient ascent or descent are very useful in practice [99].

4.1.3 *Non-Linear Problems*

Many patterns in real-world data do not exhibit linear relationships. Especially in highly complex systems such as cells, assuming linearity might not be justified [100]. Different strategies have been developed for such scenarios (depicted in Figure 4.4). The most straightforward idea is to fit another function, that is not a line, to the data (a circle or ellipsoid for the setting in Figure 4.4). This strategy is used by several algorithms [107, 108] but can have several drawbacks and often requires knowledge of the data. Most high-dimensional data sets cannot be visualized easily, leaving the choice of function classes open. Furthermore, the number of parameters to learn per dimension can increase substantially, for instance when using polynomials. This in turn can lead to problems like overfitting which will be discussed in Section 4.2.
Another approach for modeling non-linear problems is to use multiple linear classifiers and take a consensus vote of the individual classifiers in the end [109, 110]. This

approach, termed ensemble learning, works very well in practice in domains where the amount of data is limited because each "weak" classifier focuses on a part of the problem [108].
A third common strategy is to transform the data such that the problem becomes linear [111]. This strategy to handle non-linearity in data sets is most exhaustively used by kernel methods [112–114]. Instead of solving the classification problem in the original space, they employ a transformation step to the data, trying to linearize it. After the transformation step, the classification problem can be solved by a linear classifier. While the choice of the transformation function is crucial but not straightforward, the transformation often produces extremely high-dimensional spaces where linearity is either guaranteed or very likely [111]. At the same time, the data transformation can be achieved without ever computing the high-dimensional feature space through a mathematical reformulation termed kernel trick. Kernel methods were applied very successfully in various machine learning problems [106].
Each of the strategies has its advantages and disadvantages. To model a problem in the original feature space, we had to fit more complex functions but such methods are highly explainable (we can see what they learn and why predictions are made). However, most data sets cannot be investigated visually, and therefore, an appropriate function class is hard to find. At the same time, their increased flexibility requires more weights to be trained. In addition, polynomial regression models are very sensitive to outliers, reducing their capacity to generalize the model to unseen data.
Ensemble methods, on the other hand, often perform very well but when the decision boundary is highly complex, many weak classifiers are required to successfully model the problem. Therefore, they are often harder to apply in practice [106, 108].
The third option, which is based on data transformation, relies on manually designed kernel functions to simplify the problem. Furthermore, kernel methods need to store the entire kernel matrix $\in \mathbb{R}^{N\times N}$ in memory, which does not scale well when they are confronted with millions or billions of data points [99].

4.2 EVALUATING MACHINE-LEARNING MODELS

The goal of **ML** models is to be able to extract meaningful patterns from a data set in order to make predictions. The construction of a model usually involves a training (or fitting) phase after which the model can make new predictions, such as predicting expression levels of genes or human faces in images. Ideally, one would like to know how well a model performs when it is confronted with new data points that are similar to the training data. To do so, one can measure how accurately the training data points were predicted after the training phase. However, a simple algorithm that constructs a lookup table from the data would perform perfectly but still fail with high probability in a real-world scenario.
To access the model's performance on future data, the available data set is usually split into a training and test set. The test set can be used after the training phase to assess the model performance on unseen data and hence acts as an estimate of what results can be expected from the **ML** model in practice.

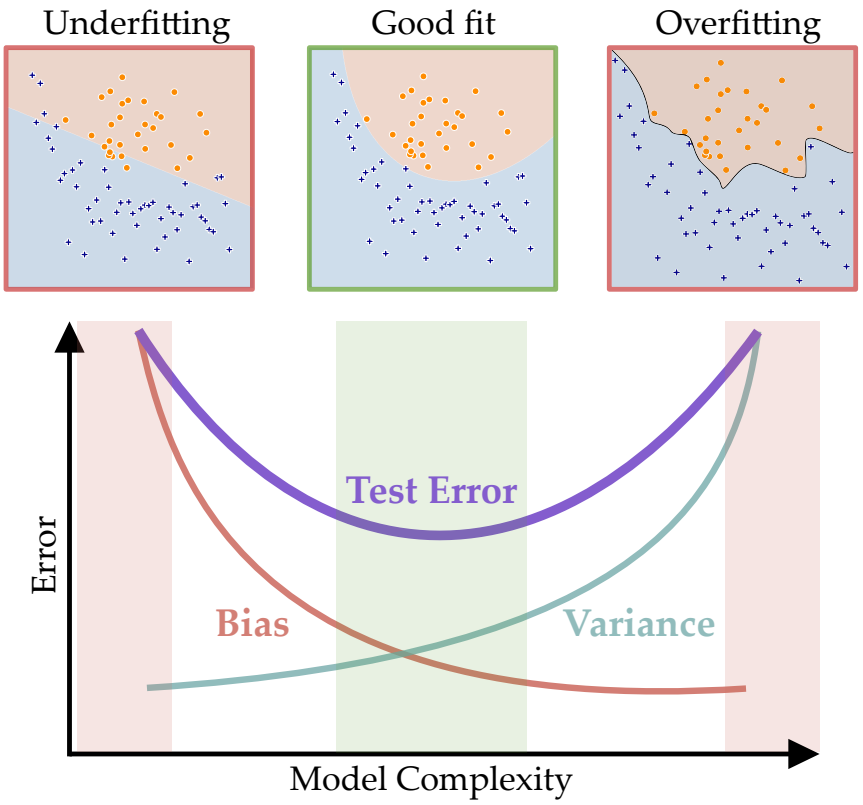


Figure 4.5: **Bias-variance tradeoff as a function of model complexity.** More complex models (those with more free parameters to optimize) tend to overfit the data (displayed in the upper right corner on a binary classification data set) while simpler models are unable to adequately model the generative process. To obtain machine learning models that work well on unseen data, a compromise between overfitting and underfitting is sought. The concepts of overfitting and underfitting are directly related to bias and variance.

4.2.1 *Bias-Variance Tradeoff*

Both, training and test error (the number of incorrectly classified data points in the respective data sets), have their own characteristics with regard to the complexity of the used machine-learning model. The complexity of a model is given by the number of parameters that are optimized during training. Highly complex models with many parameters are prone to overfitting. That means, they overestimate the data generation process and start to model outliers extensively. Very simple models, on the other hand, tend to underestimate the data generation process because they do not have the capacity to model the problem correctly. As depicted in Figure 4.5, a linear model is unable to properly classify the data due to the constraints in the underlying class of functions (a line) while a very complex model fits outliers, leading to a very "wiggly" decision surface.
The expected error on unseen data can be formally written and decomposed into bias and variance (see Section A.1 for a derivation of the bias-variance decomposi-

tion). The bias measures how well the learning algorithm captures the relations in the data while variance describes how extensively the model captures noise in the data. Ideally, **ML** models generalizing well to unseen data would have minimal bias and variance. Both, bias and variance, are directly related to the model complexity as depicted in Figure 4.5 but also to the amount of training data. The more data points available the more complex models can be trained without the risk of overfitting.
Models with minimal bias and variance are believed to be impossible and a compromise must be found [99, 106]. Finding the optimal balance between the two is usually done by observing the training and test error as a function of model complexity. When using iterative training procedures as gradient descent (described in the previous section), the training time is often used as a proximal measure of complexity and training is stopped when the test error starts rising again.
In situations where the amount of training data is scarce, model performance can often be increased by using domain knowledge. That is, by making assumptions about the data that are true in practice and that lead to fewer weights in the model. Section 4.4 introduces convolutions as an example of a method to encode basic assumptions about images into models, thereby reducing the complexity and ultimately the number of parameters to train by orders of magnitude.

4.2.2 *Performance Metrics*

An ideal machine learning model generalizes well to unseen data. That capacity is measured on a test set of data which was not shown to the model during training. The metrics, used to finally evaluate machine learning models, differ between problems. To see if model training converges (for iterative training procedures), the **loss** is often assessed. The loss describes the quantity optimized by the training algorithm. When the loss of a model starts to increase on the test set while still decreasing on the training set, overfitting occurs. The most widely used metric for multi-class problems (more than two classes) is **accuracy**. This metric simply counts the fraction of correctly predicted data points and is often more meaningful than the loss of the model. In a binary classification setting, accuracy requires the choice of a threshold when the model's output are probabilities to assign a class to a data point. Most **ML** models output a probability score for each class and data point. Hence, the choice of a cutoff or threshold is required to assess accuracy in such cases. To alleviate problems with thresholding, **area under the curve (AUC) metrics** are widely used in binary classification settings. Here, different types of errors are used as a function of the threshold, and the integral under the resulting curve serves as a performance indicator. The precision-recall curve is the most popular performance measure for binary problems. It depicts recall (how many of the positive data points were found) on the x-axis and precision (how many of these predictions are correct) on the y-axis for all possible thresholds of the model (most often 0 to 1) and measures the integral under the resulting curve. Such a metric is well suited for problems with class imbalance (one class has more data points than the other) where other metrics often fail.

4.2.3 *Cross-Validation*

Complex algorithms have many additional parameters that influence their performance in practice like regularization, convergence parameters and model complexity parameters. Such additional parameters which are not optimized during the training process are called hyper-parameters. Choosing them adequately is often not trivial, yet their optimal values highly depend on the data set used. To tune the hyper-parameters of a machine learning method, the training set is further split into a training and validation set, reducing the amount of data left for training once more. **Cross-Validation (CV)** refers to the process of iteratively training models on parts of the data while dynamically changing the validation set. In 10-fold **CV** $\frac{9}{10}$ of the training data is used to train a configuration (specific setting of hyper-parameters) of the model while the 10^{th} part is used for validation. Subsequently, nine additional models are trained, each using the next 10^{th} of the training set as validation set. Performance of the 10 models is averaged to assess the quality of the configuration. This way, the validation set can be used in parts for training. **CV** exchanges computational time for more usable training data.

4.3 NEURAL NETWORKS

An **Artificial Neural Network (ANN)** is a non-linear, supervised machine learning method that can be used for regression and classification. It can be understood as a hierarchical ensemble of logistic regression classifiers that are trained together. To simplify and visualize an **ANN** individual classifiers (or units) are thought of as a node in a network. Figure 4.7 depicts such a visualization along with the formula for the classifier. The framework for neural networks is very general and has inspired many different algorithms and variants [99].
Typically, **ANN**s are organized in hierarchical layers. Layer l receives input from the previous layer $l-1$ and the first layer receives a data point as input as depicted in Figure 4.8. Conversely, the last layer produces an output for the data point (a vector of class probabilities in classification problems or a vector of real values for regression problems). Each layer is an ensemble of individual classifiers (called units or neurons) and traditionally logistic regressions are used as units.
Intuitively, neural networks adopt to non-linear data by combining two of the ideas from Section 4.1.3 and Figure 4.4. On the one hand, each of the layers corresponds to an ensemble classifier that constructs several decision boundaries on the data set. On the other hand, each layer transforms its input data in a way that facilitates the overall learning problem. Figure 4.6 visualizes how **ANNs** solve a non-linear problem by transforming the data until the problem becomes linear. By using the output from the first ensemble learner (two logistic regression classifiers on the left), the non-linear xor-problem becomes suddenly linear because the orange points are linearly separable on the right.

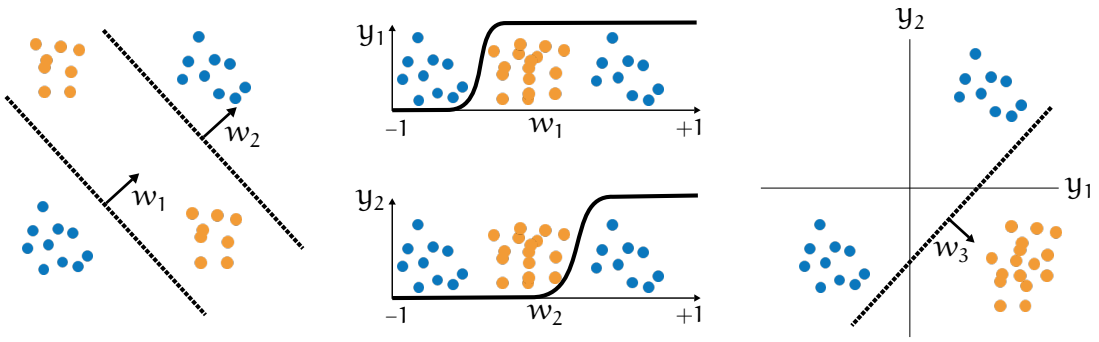


Figure 4.6: **How neural networks combine ensemble learning and data transformations to solve non-linear problems.** The results of two weak learners on the left are depicted in the middle. They transform the data until the problem becomes linear. A third weak learner (depicted on the right) then separates orange and blue points.

4.3.1 *Multiple Layers of Transformations*

As presented in Section 4.1.1, a logistic regression transforms data points by projecting them on a weight vector w and then applying a logistic function to the projection. This process is again depicted as in Figure 4.7 in a compact graphical notation along with the mathematical formulation from Section 4.1.1. Such a compact representation helps to visualize **ANN**s that wire multiple logistic regression units together. Multiple units that receive the same input but learn individual weight vectors are called **layers**. A layer of a neural network mathematically corresponds to the multiplication of a weight matrix W with the data point x or even the whole data set X. This leads to the layer-wise propagation rule:

$$H^{(l+1)} = \phi\left(W^{(l)} H^{(l)}\right) \tag{4.6}$$

where $H^{(0)} = X$. The number of columns of $W^{(l)}$ corresponds to the number of units for that layer, and the number of rows corresponds to the dimensions of X (or the input to layer l for that matter).
Multiple layers can be stacked on top of each other by using the output of layer l —

$$\begin{aligned} y &= \phi\Big(\sum_{i=1}^{p} w_i x_i\Big) \\ &= \phi\left(w^T x\right) \end{aligned}$$

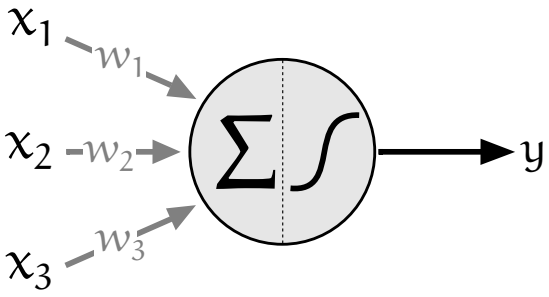


Figure 4.7: **Formula and graphical representation of an ANN unit.** A unit in an **ANN** receives an input vector (x), projects it on the weight vector w, and then applies a non-linear function on the projection. The left depicts the formula, corresponding to the graphical representation on the right.

denoted $H^{(l)}$ — as input to layer $l+1$. Figure 4.8 depicts such a neural network with different unit sizes in more detail. Layers that are neither input nor output layers are referred to as **hidden layers** (marked in green and red). Their role is to consecutively transform the data such that it becomes linear at the output layer.

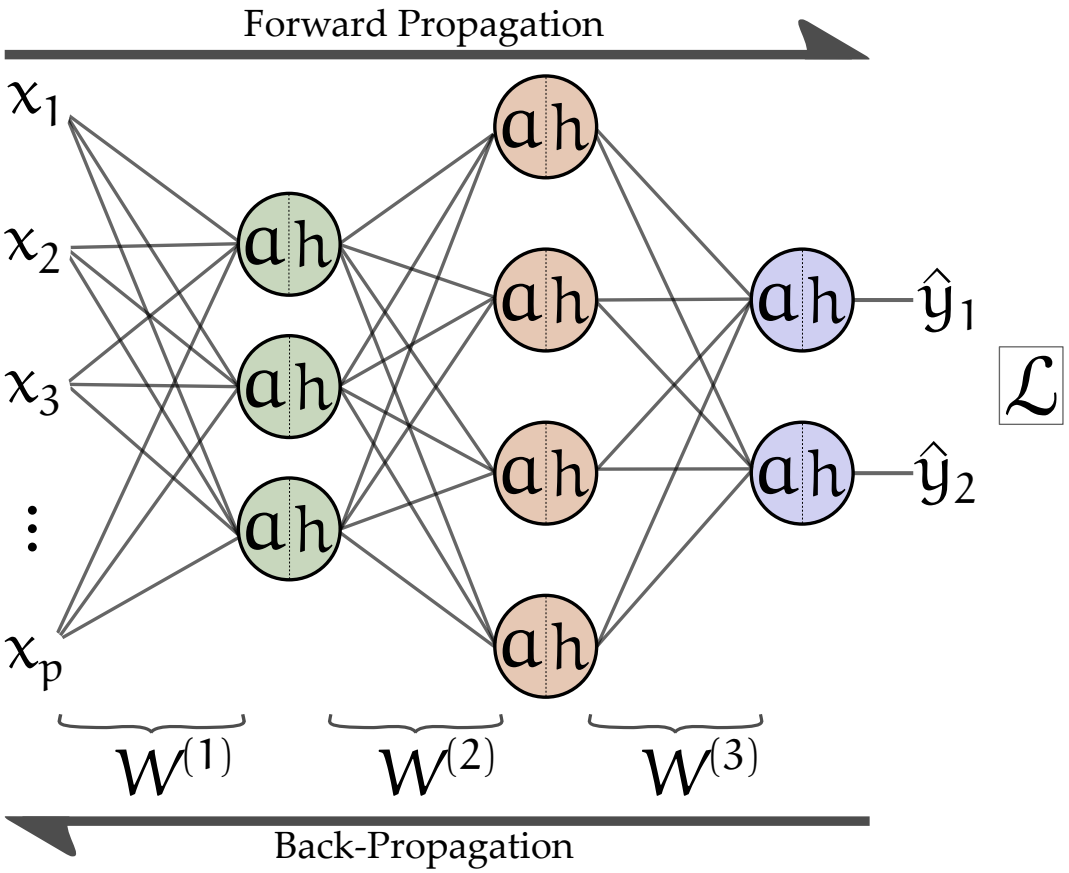


Figure 4.8: **A complete neural network composed of two hidden layers with** 3 **and** 4 **units each.** The input vector $x \in \mathbb{R}^p$ is "clamped" to the input units. It then flows through three layers of transformations until reaching the output layer (in blue). The depicted **ANN** transforms x to a two-dimensional output vector. The weights can be summarized as a weight matrix $W^{(l)}$. At each unit a denotes the projection $w^T x$ and $h = \phi(a)$ corresponds to the unit's output.

The **activation function** ϕ of an **ANN** does not have to be a logistic function and does not have to be the same across all layers. In fact, logistic functions are not very well suited for deep neural networks (those with multiple hidden layers) because they saturate at 1 [115]. The most widely used activation function is the **rectified linear unit (ReLU)** function [99, 116, 117], depicted in Figure 4.9 alongside other popular activation functions. Importantly, the activation functions are the only introduction of non-linearity into the data. Without them the whole formulation of a neural network collapses to recurrent applications of weight matrices (and therefore linear functions). This could be written as only one linear transformation, yielding a simple ensemble classifier without hierarchical structure.

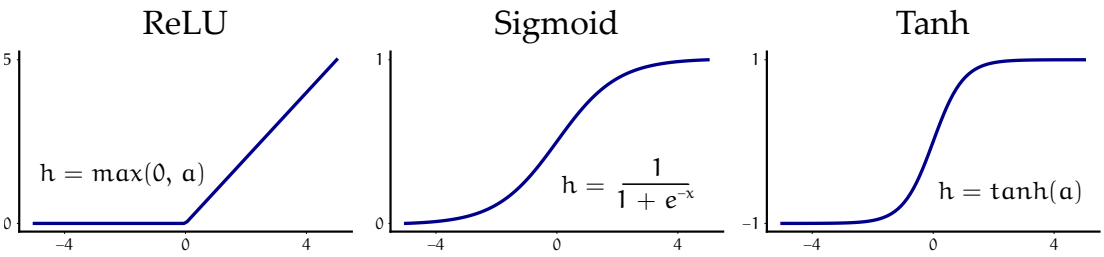


Figure 4.9: **Popular choices for activation functions.** The **ReLU** function is the currently most widely used activation function. The non-linearity at 0 is explicitly treated in implementations and set to zero. The sigmoidal activation function is often used for the output layer in binary classification. The hyperbolic tangent function (tanh) is similar to the sigmoidal function but sometimes faster to evaluate.

Similar to most machine learning models, neural networks are trained through the optimization of a performance measure P as defined in Section 4.1. In the context of deep learning, the performance measure is most often called **loss function** with popular choices for the loss function being the **mean-squared error (MSE)** or the cross-entropy function [99] given by:

$$\mathbb{L} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{4.7}$$

Training refers to the process of starting with a random initial guess for the weights and iteratively minimizing the loss function through gradient descent, similar to the procedure for training a logistic regression classifier in Section 4.1.1. For that, **ANN**s have to be fully differentiable, limiting the choice of activation functions. For a single logistic regression training is relatively simple. For an entire network of such classification units, a key question is how the loss $\mathbb{L}$ can be used to compute partial derivatives of all the weights in order for the **Gradient Descent (GD)** algorithm to work. The answer is an algorithm called backpropagation. This algorithm is used to train most neural networks (including those discussed throughout the thesis but also most unsupervised variants such as generative adversarial networks, variational autoencoders, transformer models and other state-of-the-art neural networks). It is highly efficient and explained in more detail in the next section.

4.3.2 *The Backpropagation Algorithm*

Neural networks are almost exclusively trained with a form of the **GD** algorithm. It regards the network as a function $f(x;\Theta)$ that is differentiable and is parametrized by $\Theta = \{W^{(1)}, W^{(2)}, \dots W^{(l)}\}$. As seen in Section 4.1.2, **GD** minimizes the loss function in an iterative manner. For that, it requires the partial derivatives of the loss with respect to the model parameters (see Equation 4.5).

The **backpropagation** algorithm computes the partial derivatives of the loss function with respect to the individual weights in a neural network denoted by:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}}$$

in an efficient manner. It makes use of the chain rule of calculus, thereby distributing the gradient from the output layer to the last hidden layer and from there further down the hierarchy. Let $y = g(x)$ and $z = f(y) = f(g(x))$. Then the chain rule states that:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x} \tag{4.8}$$

The calculations of the neural network from Figure 4.8 can be written by using the layer-wise propagation rule from Equation 4.6 as:

$$\hat{y} = \phi\Big(W^{(3)}\phi\big(W^{(2)}\phi(W^{(1)}X)\big)\Big) \tag{4.9}$$

which is a nested application of functions.
The key idea in backpropagation is to compute the derivatives starting from the last layer and to express them in terms of derivatives that were already computed in a higher layer by using the chain rule.
In more detail, backpropagation starts by computing the derivative of the loss function with respect to the input for every unit in the network. Those derivatives, denoted δ, can be expressed solely in terms of the above layers when using the chain rule. Therefore they can be computed in a reversed layer-wise manner starting from the output and going back to the input layer.
The derivative of the loss $\mathcal{L}$ for a unit i in layer l with respect to its input can be regarded as the amount of error that the unit contributes to the overall loss. It can be formulated as:

$$\delta_i^{(l)} = \frac{\partial \mathcal{L}}{\partial a_i^{(l)}} = \frac{\partial h_i^{(l)}}{\partial a_i^{(l)}}\frac{\partial \mathcal{L}}{\partial h_i^{(l)}} = \phi'(a_i^{(l)})\sum_{j=1}^{|l|} w_{ji}^{(l+1)}\delta_j^{(l+1)} \tag{4.10}$$

where $a = w^T x$ denotes the projection of x on w, $h = \phi(a)$ the output of a unit and ϕ' corresponds to the first-order derivative of the activation function ϕ. From Equation 4.10 it becomes immediately clear that the activation function ϕ needs to be fully differentiable. The derivative of the loss with respect to the output of the **ANN** depends on the choice of the loss function, for instance $\phi'(y_i - \hat{y}_i)$ for the mean-squared error loss function where $\hat{y}_i = f(x_i)$ denotes the network output for data point x_i.
Once all δ values are computed in the network, the partial derivative of the loss with respect to a weight can be written as:

$$\frac{\partial \mathcal{L}}{w_{ij}^{(l)}} = h_i^{(l-1)}\delta_j^{(l)}. \tag{4.11}$$

We have now seen how the partial derivatives of the loss with respect to each of the weights can be computed efficiently. It requires additional storage for the δ values at

each unit but involves only pointwise products and can therefore execute very well on SIMD architectures, such as **Graphics Processing Units (GPUs)** or **Tensor Processing Units (TPUs)**. The derivatives for the activation functions are usually simple and require linear computation time.
Backpropagation works very well in practice and can be scaled up to millions or billions of data points, only increasing training time but not memory requirements. This is a fundamental difference to other popular non-linear machine learning algorithms, such as kernel machines. It is believed to be the main reason why deep neural networks have shown great success in modern large-scale learning problems [99, 118].

4.4 CONVOLUTIONAL ARCHITECTURES

A major drawback of the standard fully connected **ANN** presented so far is the high number of parameters that have to be learned when the input is high-dimensional (p is large). In such a scenario, the first layer contains at least p weights even when only using one unit in the first hidden layer. This behavior is often unacceptable because it yields a highly complex model with the associated disadvantages, such as overfitting (see Section 4.2.1). Especially when working with images which are inherently high-dimensional (the number of pixels), fully connected neural networks are unsuited. As introduced in Section 4.2, a common strategy in such cases is the incorporation of domain knowledge into the machine learning task to reduce the variance of the model. The edges and their weights in an **ANN** can be wired arbitrarily, allowing for very flexible connection patterns across layers.
In the case of images, patterns can be located in any area of the image and are usually only spanning across parts of the image. Therefore, the local neighborhood of a pixel is more informative for characterizing patterns than distantly located pixels.
These properties can be incorporated in a neural network by constructing units that act as feature detectors. By using the same few weights at many locations of the image, genomic sequence, or audio sample the unit has to find a pattern in the data rather than partitioning the entire data space. In this process, referred to as weight-sharing, a unit that slides over the input data point corresponds to a local feature detector and its activation peaks when the pattern of interest is found in the data. This corresponds to a common pre-processing strategy for image data that was used before the era of deep learning, where Sobel-Feldman filters [119] or other hand-crafted feature detectors were used to find interesting patterns in the images. The location and intensity of the detected features were used as input for a machine learning algorithm [99, 120].
The sliding window approach can be mathematically written as a convolution operation widely used in image or speech processing.

4.4.1 *Convolutions & The Convolution Theorem*

The convolution operation is applied widely in the natural sciences but has gained a slightly different meaning in machine learning than in other domains [99]. In machine learning a convolution applies a small discrete function w to a larger series of

measurements x. w is often referred to as the **kernel**, x is the input and the result of the convolution operation is denoted as the **feature map**. Typically, the kernel function is represented as a matrix and the entries in w are learned via backpropagation while x denotes a data point.
The one-dimensional convolution operation can be written as:

$$c(i) = (x * w)(i) = \sum_{j}^{k} x(i)w(i-j) \tag{4.12}$$

but can be extended to sliding over two dimensions simultaneously for the use of images or three dimensions for applications to video and other temporal data.
The convolution operation is characterized by the size of the kernel k. Figure 4.10 depicts a one-dimensional convolution operation and the corresponding architecture of a neural network. Interestingly, convolutions are pointwise products of input and

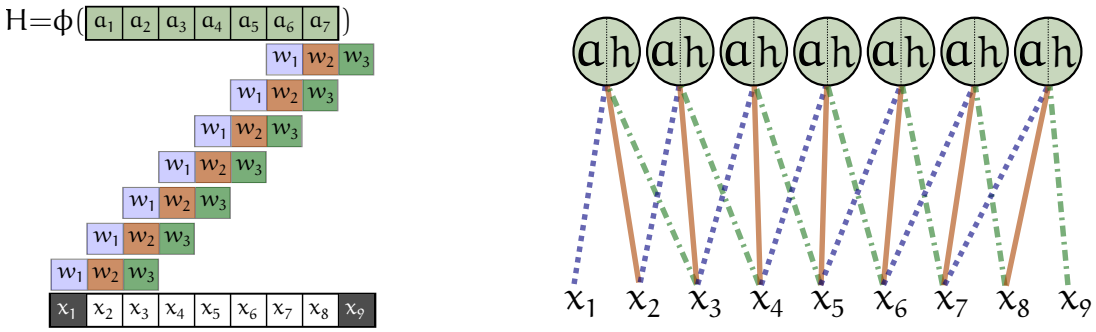


Figure 4.10: **Example of 1D-convolution between a 9-dimensional input vector and a 3-dimensional kernel.** The input vector $x \in \mathbb{R}^9$ is convoluted with a kernel $w \in \mathbb{R}^3$. Left: The sliding window approach uses the same weights for the whole sequence x and forms an activation vector a that indicates "hits" of the pattern characterized by w. Right: The same convolution operation as a neural network with sparse connections and weight-sharing. The edge color indicates the weight.

kernel matrices in the frequency space. The frequency space is a common representation of high-dimensional data (typically images or audio) as a set of frequencies. It is computed through the well-known Fourier transform. Without going further into details of the Fourier transform, the convolution theorem states that the convolution depicted in Figure 4.10 can be written as a pointwise multiplication in the frequency space. Let $\mathcal{F}(x)$ denote the Fourier transform. Then the convolution theorem states that:

$$(x * w) = \mathcal{F}^{-1}\big(\mathcal{F}(x) \cdot \mathcal{F}(w)\big) \tag{4.13}$$

For neural networks, this theorem not only has runtime advantages when chaining multiple convolution operations (through first transforming x to the frequency domain, applying multiple convolution operations and then re-transforming x to the original domain) but can also help to extent convolutional architectures to domains where Fourier transforms are defined but the convolution operation is not. An example of such a domain are graphs which have no natural definition of a sliding-window operator but a definition of a convolution operator, introduced in Section 4.5.2.

4.4.2 *Convolutional Neural Networks*

A **Convolutional Neural Network (CNN)** uses hierarchies of convolutional layers [120]. The convolution operation is implemented through the construction of a weight matrix with the same values repeated in a specific order (called Toeplitz matrix) which is multiplied by the input. It is then followed by a pooling operation where the result of the convolution is shortened. Max-pooling, for instance, simply takes the maximum value of 2 or 4 adjacent values and discards the others. Pooling helps to shrink the size of the convolution results, thereby reducing the size of the layer output compared to the input. The ensemble of convolution and pooling characterizes a convolutional layer. Typically, several convolutional layers are followed by a variable number of fully connected layers introduced in the previous section.
With the consecutive stacking of convolutional layers, a **CNN** introduces another bias to the model. It not only assumes that the location of a pattern is less important (translation equivariance) but also that patterns are organized hierarchically. A human face, for example, is made up of eyes and a mouth, which in turn have several characteristics and are composed of simple edges and strokes in the image. **CNN**s have been successfully used for image modeling tasks [97, 121] and have by now become the de facto standard for image recognition problems [118]. Indeed, several studies have found that the human brain processes images in a similar hierarchical fashion [118, 122, 123].
The huge success of **CNNs** raises the question if convolutional architectures can be applied to other domains where the input does not correspond to linear sequences or regularly arranged pixels. Biological networks, for instance, are presumably organized in modules that in turn consist of sub-modules and ultimately protein complexes. However, how to define a convolution on a graph is not clear. The degree of nodes (the number of nodes a node is connected with) differs substantially between proteins [124–126]. In addition, there is no natural order of neighbors of a node [127, 128] so nodes cannot be treated as a sequence.

4.5 GRAPH DEEP LEARNING

Before going into the generalization of convolutions to graphs, we need basic knowledge about graph theory. We will see how to formalize a graph as sets of nodes and edges and how random walks provide a stochastic framework to examine neighborhoods of nodes. From there, we will define convolutions on graphs and see how we can simplify them to work in the deep neural network framework.

4.5.1 *Graph Theory*

Graphs are one of the most fundamental data structures in computer science and the core of many real-world applications [129].
We define a graph $G = (V, E)$ as a pair of vertices (V) and edges (E). An edge connects two vertices i and j can optionally have a weight $w_{ij} \in \mathbb{R}$ attached to it. Edges can be directed or undirected. Vertices are also called nodes and throughout the thesis those

two will be used interchangeably. A node i can be associated with attributes in the form of a vector $x_i \in \mathbb{R}^p$.
Mathematically graphs are often represented by their *adjacency matrix* $A \in (0,1)^{N \times N}$. An entry in $A_{ij} = 1$ means that nodes i and j are connected while a 0 represents no connections. If the edges in G are undirected A will be symmetric. If the edges are weighted, A contains the weight instead of the boolean indicators for connections. The degree of a node denotes the number of connections that the node has with other nodes. The degree matrix $D = diag(\sum_j A_{ij})$ contains the degree of each node on the diagonal while all other entries are 0. Apart from the degree, many more metrics exist to describe the position of a node inside the network. Many of them were originally defined for social networks but are applied in different life sciences as well. *Betweenness centrality* of a node i denotes the number of shortest paths in the graph that pass i [130]. The *core* (sometimes referred to as k-shell) is another measure of centrality and is determined iteratively by removing isolated nodes from the network. The longer a node "survives" those iterations, the more central it is in the network and the higher is its core [131].
In many real-world cases nodes have certain properties that describe them. For instance, a social network, where people are represented by nodes, might contain other characteristics of the individuals as vectors. Such a vector, in which each dimension represents a node of the graph, is referred to as a signal $x \in \mathbb{R}^N$ on the graph and the field of **Graph Signal Processing (GSP)** is concerned with analyzing the node features with respect to the graph. Multiplying the adjacency matrix A and the signal x, for instance, smooths the signal over the graph by aggregating the signal from neighboring nodes.

Spectral Graph Theory

Similar to the adjacency matrix, the laplacian matrix L of a graph is defined as:

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}. \tag{4.14}$$

The graph laplacian is a symmetric matrix when the graph is undirected. Therefore, it has real-valued non-negative eigenvalues. When taking the intuition of a random walker from above the eigenvectors of the graph laplacian denote those configurations where making a step would change the distribution of the walker only by a constant. In other words, the signal received by a node is equivalent to the signal of the node itself. The graph laplacian deducts the incoming signal (the off-diagonal entries contain negative values) while scaling the node signal by the degree. Hence, the eigenvectors of the graph laplacian denote frequencies of the graph.
Formally, the eigenvectors associated with the eigenvalues form an orthonormal basis and the eigenvalues are nonnegative real values. The eigenvalues of the laplacian have several interesting properties. For instance, the smallest eigenvalue is 0 when the graph is connected. If not, it denotes the number of components in the graph. The second-smallest eigenvalue solves the minimization problem

$$\lambda_2 = \min_x \frac{x^T L x}{x^T x} = \sum_{(i,j) \in E} (x_i - x_j)^2$$

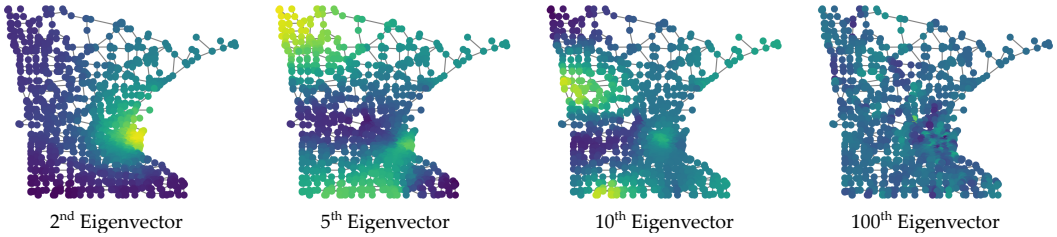


Figure 4.11: **Different eigenvectors on the Minnesota road graph.** The **GFT** produces eigenvectors sorted by their corresponding eigenvalues. The first eigenvectors capture low frequencies and smooth patterns in a graph signal. The later eigenvalues capture higher frequencies, analogous to the classical Fourier transform [138].

for the eigenvector x and its corresponding eigenvalue λ_2. This yields a partition of the graph in two subgraphs and therefore describes the lowest frequency of G. In a similar fashion, the larger eigenvectors describe higher frequencies in G. This is visualized on the Minnesota road network graph where edges in the graph are roads connecting landmarks in Figure 4.11.
We have now seen how solving the eigen-decomposition of G produces a basis of frequencies of the graph G. This decomposition is called the **graph Fourier transform (GFT)** and works analogous to the discrete Fourier transform.

4.5.2 *Convolutions on Graphs*

Since the successes of **CNN**s on image processing problems, efforts have been made to generalize them to non-regular grids, such as graphs [128, 132–137]. An image can be regarded as a graph where each pixel corresponds to a node and edges denote adjacent pixels. A biological network, however, does not have such a regular pattern of connectivity, making it hard to formulate a convolution operator over such a graph. Two main assumptions make **CNN**s so successful on many machine learning problems: First, transitional equivariance means that the exact location of a pattern in the data is not important but the detection of the pattern is and second, local support means that the detector of the pattern is much smaller than the input dimensions. Together, those two assumptions allow for weight-sharing and make the number of parameters to train independent of the dimensionality of the input [139]. To extend convolutions to graphs, one has to construct an operator that preserves those assumptions.
Graph convolutions can generally be divided in spatial or spectral approaches. The former tries to find a way to construct local neighborhoods of a graph, transform them into matrices and then apply convolutions and pooling similar to standard **CNN** architectures [127]. The latter makes use of spectral graph theory and the convolution theorem to formulate a convolution operation on graphs.
In Section 4.4.1 we learned how a convolution becomes a pointwise product in the Fourier space of a function. This is possible because convolutions are linear operators that diagonalize in the Fourier domain [132]. As introduced in the previous section, the eigenvectors of the graph laplacian L (4.14) form a Fourier basis and the **GFT**

therefore corresponds to an eigen-decomposition of L that transforms the graph to a frequency domain over the nodes [140]. The **GFT** is defined as:

$$L = U \Lambda U^T \tag{4.15}$$

where U is the matrix of eigenvectors and Λ the diagonal matrix with the corresponding eigenvalues. Equation 4.15 can be solved using eigen-decomposition or **Singular Value Decomposition (SVD)**. Let $x \in \mathbb{R}^N$ be a signal on a graph G, corresponding to a scalar value per node. x can be convoluted with a filter $w \in \mathbb{R}^N$ in the frequency domain of the graph by:

$$x * w = U\Big((U^T x) \cdot (U^T w)\Big) \tag{4.16}$$

In this form the kernel w of a graph convolution learns to recognize patterns of the nodes in the spectral domain. The filter w is theoretically translation equivariant due to the convolution theorem. However, w has to be N-dimensional and therefore does not operate locally on regions of the graph, unlike a convolution on images or speech. Furthermore, the number of parameters depends on the number of nodes in the network. If the machine learning task is the classification of nodes, the number of parameters for a single graph convolution is at least as high as the number of data points.
To construct localized filters that operate only in a small region of the graph at a time, polynomials with local support can be used [128, 133]. Here, a polynomial (usually recursively defined Chebychev polynomials) is used to construct a filter in the vertex domain (prior to the transformation to the spectral domain). The degree of the polynomial defines a local support around a graph node and is constructed using only the nodes around a center node i. More specifically, a Chebychev polynomial of degree K around a node i is guaranteed to have support over vertices that are at most K hops apart from node i [141].
A Chebychev polynomial is recursively defined as:

$$T_k(L) = 2LT_{k-1} - T_{k-2}(L) \tag{4.17}$$

with the anchors for recursion being $T_0(L) = 1$ and $T_1(L) = L$. Using such polynomials of the laplacian, a graph convolution can be approximated by:

$$x * w \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L}) x \tag{4.18}$$

where $\theta_k \in \mathbb{R}^K$ is a vector of Chebychev coefficients and thus the learnable parameters of a graph convolution. $\tilde{L} = \frac{2L}{\lambda_{max} - I}$ denotes a scaled version of the laplacian where λ_{max} corresponds to the largest eigenvalue of L.

tThe Chebychev polynomial from Equation 4.18 is an approximation to the original spectral convolution operator that is localized. That is, the number of parameters depends on K and no longer on the number of nodes or features [128]. Furthermore, the **SVD** of the laplacian is no longer needed which makes the method computationally much more efficient and renders it usable for large graphs.

4.5.3 *Graph Convolutional Networks*

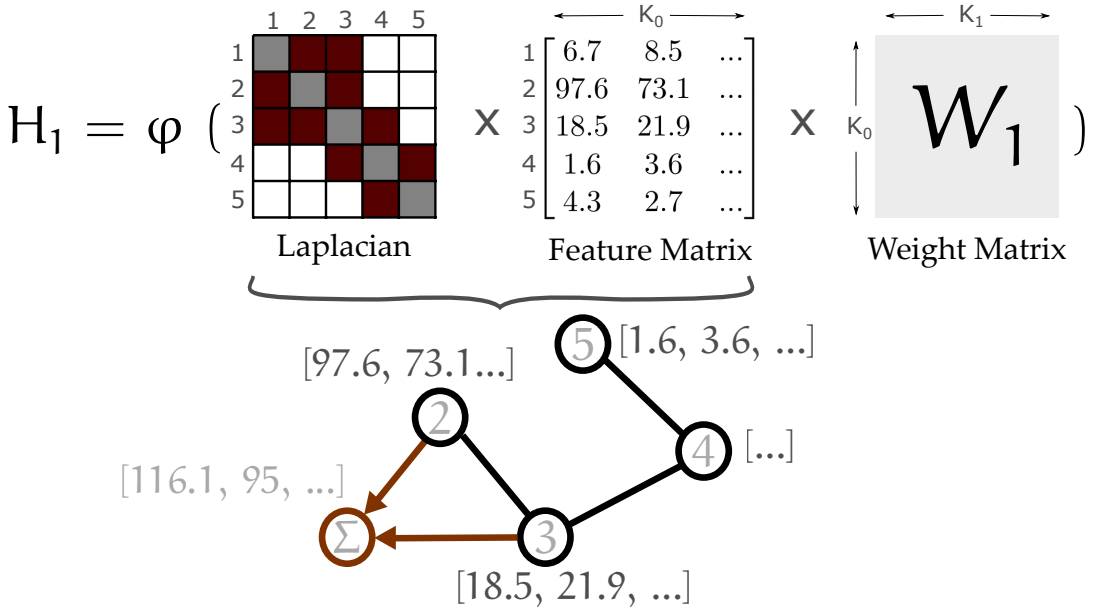


Figure 4.12: **Visualization of a graph convolutional layer as used by Graph Convolutional Networks [133].** The first multiplication of L and X conducts a one-step random walk for every feature independently. The result is linearly transformed by a weight matrix. A **GCN** with one layer already greatly outperforms a fully connected network [142].

An observation from studying **CNN**s shows that higher layers automatically aggregate information from lower layers beyond the reach of the filters (visualized in Figure 4.13) even when no pooling is used [99]. This is because the result of a convolution contains information about nearby features, and therefore the next convolutional layer will "see" larger neighborhoods. The same holds true for graph convolutions and can be used to further simplify their formulation. By restricting $K = 1$ only the first-order neighborhood of a node (direct neighbors only) is considered [133] and larger neighborhoods are accounted for through multiple graph convolutional layers [142]. Furthermore, by simply setting λ_{max} to 2 the layer-wise propagation rule of graph convolutional networks can be written independently of the polynomial formulation. Neural networks are expected to learn the scaling of the laplacian automatically through a bias term [133], justifying the approximation of λ_{max}. A graph convolutional layer can then be defined through a graph convolution and a nonlinear activation function. The convolution operator from Equation 4.18 was defined for only one-dimensional features but can easily be extended to multidimensional features. The final simplified propagation rule of a **Graph Convolutional Network** (**GCN**) [133] becomes:

$$H^{(l+1)} = \phi\left(\tilde{L}H^{(l)}W\right) \tag{4.19}$$

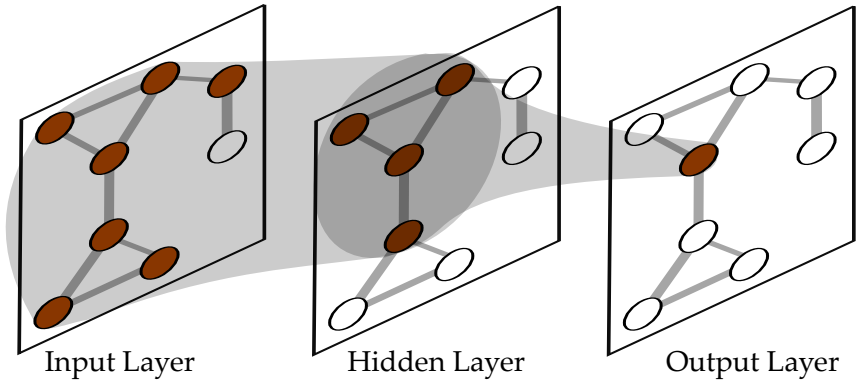


Figure 4.13: **Size of the receptive field in CNNs and Graph Convolutional Networks (GCNs).** One unit in the output layer receives information from all neighboring nodes in the layer below (denoted hidden layer). The same effect holds for the input layer. Almost the entire input layer is aggregated in a single output unit, making the use of pooling not strictly necessary. This has to be taken into consideration when designing **GCNs**.

where $H^{(0)} = X \in \mathbb{R}^{N \times p}$ is a p-dimensional feature matrix and $\tilde{L}$ denotes the laplacian from Equation 4.14 but replaces A with $\tilde{A} = A + I$ (added self-connections). This has the effect of aggregating the feature values of the node itself into the graph convolution.
The simplified layer-wise propagation rule strongly resembles a fully connected propagation rule from 4.6 only that it is multiplied by a normalized version of the adjacency matrix A. When interpreting the node features X as independent probability distributions over the vertices, a **GCN** performs a one-step random walk prior to a fully connected neural network layer. This procedure, known as laplacian smoothing, already has broad applications and was shown to be successful also outside the neural network formulation [69, 142].
A **GCN** stacks multiple layers of graph convolutions. In contrast to **CNN**s, no fully connected layers are used at the end [133]. Different architectures for **GCN**s have been proposed within the last years [136, 142], some of them implementing strategies for pooling [132, 139, 143]. However, pooling is not required to reduce the number of parameters in the graph setting and not conclusively shown to yield a significant benefit for **GCN**s.

Training Graph Convolutional Networks

In Section 4.3.2 we saw how neural networks can be trained with backpropagation and gradient descent to minimize the classification error of some given data set X with respect to the labels y. For the **GCN** model training, we are confronted with a graph from which it is hard to extract individual nodes for classification. Usually, only a small fraction of nodes carry a label while the majority of nodes are unknown. The classification goal of a **GCN** is to label the unknown nodes. Therefore, the loss function is computed only for the known vertices. This is a form of semi-supervised learning because the unknown nodes still influence the loss of the known ones through

aggregation in the graph convolutions [133]. Semi-supervised learning is a hybrid of supervised and unsupervised learning (depicted in Figure 4.1) where unlabeled data points are used in combination with labeled ones in the learning procedure.

4.6 INTERPRETING NEURAL NETWORK DECISIONS

Supervised machine learning is a powerful concept that enabled incredible progress in multiple sciences [144, 145]. In research it is now more and more used as a tool to gain insights into processes that are not fully understood [145–147].
Hence, it is often crucial that the underlying algorithms are somewhat transparent in their decision-making process [148–151], giving rise to the field of explainable or interpretable machine learning (nowadays often termed XAI for e**X**plainable **A**rtificial **I**ntelligence). Interpretable or explainable machine learning refers to finding models that allow users to understand what the model learned and why a certain prediction was made. Though interpretability is a broad concept, it is commonly defined as letting a user gain knowledge about the predictions made by the model [152]. As a result, interpretation depends highly on the use case that can range from verification of successful training [150, 153, 154] to gaining knowledge about scientific problems [155–157].
Interpretability can help to establish trust in a model, especially when there is no gold standard verification available as it is often the case in biology. When a model can recapitulate some of the previously identified mechanisms that connect input and output, there is a good chance that successful learning is not only based on biases and artifacts in the data but rather reflects true underlying relationships.

4.6.1 *Interpretable Machine Learning*

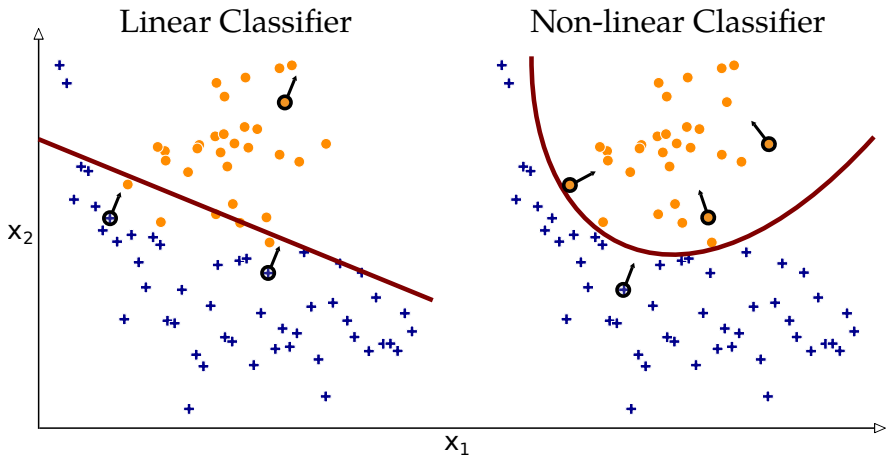


Figure 4.14: **Explanations of data points for linear and non-linear classification boundaries.** The linear case yields the same explanations for all data points because they point in the same direction away from the decision boundary. In the non-linear case explanations can be very different for different data points due to the complex decision boundary. (Figure inspired by [151])

Some machine learning models are intrinsically interpretable (such as most linear models, decision trees and small rule-based systems). This is because the model learns a coefficient for each of the features and the features with largest coefficients are the most important to the model. The interpretability comes from the model directly operating in the feature input space and the choice of an explicit function class (linear functions). Furthermore, the decision boundary of a linear classifier gives the same importance to each feature for every single data point [151].
This changes when considering non-linear models. Such models typically do not operate directly in the feature space but transform the data as pointed out in Section 4.1.3. As a consequence, each data point can have its own feature importance when using non-linear models, visualized in Figure 4.14.
Unfortunately, complex problems usually require advanced non-linear models. Especially in deep neural networks individual weights no longer carry any meaning. As a result, non-linear methods in general but neural networks in particular have been labeled black boxes [158, 159]. Many different methodologies fall under the hood of explainable machine learning [152, 159], and many of them are applicable to **ANNs**.
This thesis focuses on so-called *post-hoc* interpretation. The goal here is to query a trained model on what it has learned about the data with respect to the problem to solve.
Post-hoc interpretation of neural networks can be roughly divided into two different processes. Model-centric (sometimes called dataset-level) interpretation aims to find the features that are most important for the model with respect to the classification or regression problem that the model tries to solve. In **CNNs** model-centric interpretation often refers to the visualization of convolutional filters by various techniques [160–162].
Attribution methods (or prediction-level interpretation approaches), on the other hand, find important features for the classification of a specific data point. While for linear models the two types of interpretation are equivalent (see Figure 4.14 for a visual intuition), this is not the case for non-linear models.
When classifying cancer vs. non-cancer genes, for instance, it is highly important to know why a certain gene or data point was classified as a cancer gene. Model-centric feature interpretation would probably not yield much biological insight because every feature might be important for some genes while no single feature dominates classification. Prediction-level interpretation, however, can overcome this limitation and directly give information about why individual decisions were made.
Different attribution methods have been proposed within the last years [151, 158, 160, 163, 164], some of them being specific to **ANNs** while others are model-agnostic [164, 165].

4.6.2 *Strategies for Prediction-Level Interpretation*

Prediction-level interpretability methods can be grouped in three different approaches: perturbation-based, gradient-based and local linear methods. Each of the three strategies has its own advantages and drawbacks or is only applicable to a subset of **ML** models.

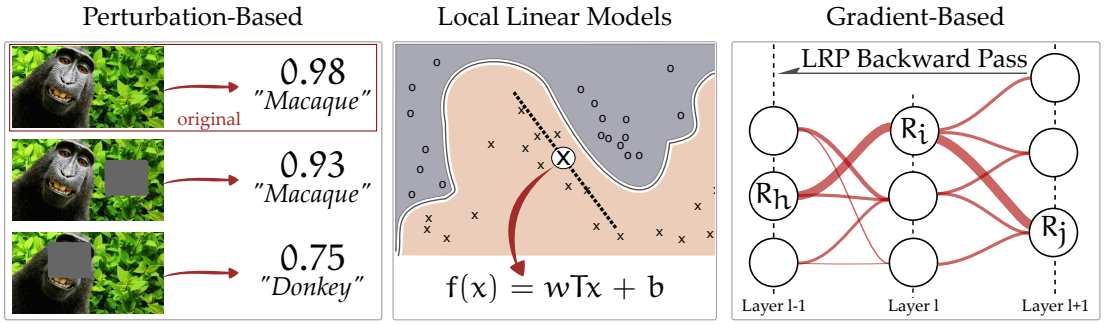


Figure 4.15: **Different Approaches for Prediction-Level Explanations. (Left)** Perturbation methods such as Occlusion hide parts of the input and assess the output probability of the trained model to identify regions of the input that were important for classification. **(Middle)** Local linear models can be trained on a subset of data points close to the point of interest. Such models are inherently interpretable and locally faithful to the original model. **(Right)** Gradient-based methods exploit differentiable models such as **ANNs** and apply a modified version of the back-propagation algorithm to identify relevant inputs. (Panel adapted from [166])

Perturbation-Based Interpretability

Perturbation-based interpretation methods attempt to change the input data point in various ways and observe the classification outcome. The occlusion or gray-box method [160] covers parts of a scene systematically and assesses the change in the prediction outcome. The Shapley value method [165] borrows random data points from the training set and inserts a feature value from the data point of interest into the borrowed data point.
The general methodology of perturbation-based interpretation methods is similar to general feature selection where a model is trained on only a subset of features to see which features are most informative for the classification task [108]. The only difference is that the models are not retrained but only specific data points are modified to hide some of the input features (occlusion) or to replace one feature with the value from another data point (Shapley values). Perturbation methods are often slow, especially for high-dimensional feature spaces [167], but do not make any further assumptions on the models.

Local Linear Models

Another strategy for the interpretation of non-linear **ML** models is the construction of a local linear model that approximates the more complex non-linear one at the point and surroundings of the data point of interest [164]. Local linear models are similar to perturbation methods, but instead of masking parts of the input, they query the model about data points close to the data point of interest. The different outputs of data in close proximity are used to construct a linear model that approximates the original one. The new local model is faithful to the original model in the surroundings of the data point of interest but linear and hence interpretable. LIME [164] for instance uses the coefficients of the linear model as feature importance of the original one.

Local linear models are also very similar to gradient-based methods because they basically estimate the gradient of the non-linear decision function at a specific point of interest. However, local linear models are model-agnostic and also work with non-differentiable models.

A downside of local linear models is the construction of a separate model to explain the decisions of the original one. It is not clear if the computed explanation corresponds to the actual reason for the classification or if the gradients of the complex and linear model point in the same direction by chance. If many explanations exist for a data point, local linear models might not produce a faithful one.

Gradient-Based Interpretability

All neural networks trained through backpropagation are fully differentiable, as discussed in Section 4.3.2. This property can be exploited by interpretation methods. Gradient-based interpretation methods attempt to redistribute the output of a specific data point back to the input space, similar to the backpropagation algorithm. In fact, early approaches used the backpropagation algorithm directly to compute the gradients of the input units of an **ANN** [168]. However, this method informs on how the input data should change in order to modify the output class but does not explain why the model assigned a specific class to the data point [166]. Thus, many more approaches for differentiable models such as **ANNs** were developed over the last years [151, 158, 163, 169, 170] with some methods being equivalent to others [171, 172]. In this thesis **Layer-wise relevance propagation** was used as a gradient-based interpretation method of choice. A more detailed explanation of **Layer-wise relevance propagation (LRP)** follows in the next section.

4.6.3 *Layer-Wise Relevance Propagation*

As mentioned in the previous section, **LRP** is an attribution method for interpreting non-linear models such as **ANNs** or **CNNs**, which exploits the differential architecture of such models [158]. To understand the contribution of a single feature of a data point x_i to the model prediction $f(x_i)$ by a classifier f, **LRP** attempts to decompose $f(x_i)$ into a sum of *relevance scores* R_d, corresponding to input dimensions. Therefore, the sum of all relevance scores approximately corresponds to the classification outcome:

$$f(x) \approx \sum_d R_d \tag{4.20}$$

The interpretation of relevance scores is that $R_d < 0$ contributes evidence against the structure that the classifier has learned, while $R_d > 0$ contributes evidence for the presence of such structure and $R_d \approx 0$ contributes no evidence to it. **LRP** redistributes the relevance scores from the output layer of the model back to the input units by applying a layer-wise propagation rule iteratively, analogously to the backpropagation algorithm (see Figure 4.8 and Section 4.3.2).
The general idea behind this algorithm is that the input domain is meaningful to a human user and the decomposition of the relevance therefore allows to determine

why a classification was made.
The total amount of relevance at each layer remains the same, leading to the *conservation property* defined as:

$$\sum_d R_d^{input} = ... \sum_i R_i^l = \sum_j R_j^{l+1} = ... = f(x). \tag{4.21}$$

This process is depicted in Figure 4.16 and can be imagined as a flow of relevance through the network. Edges with high weights associated to them will have higher amounts of relevance flowing through them, compared to edges with small weights, thereby reflecting the learned architecture of the network.
Exploiting the chain rule of calculus (Equation 4.8), the layer-wise relevance propagation rule can be formulated as:

$$R_i^{(l)} = \sum_j \frac{h_i w_{ij}}{\sum_i h_i w_{ij}} R_j^{(l+1)} \tag{4.22}$$

where R represents the relevance of node i and j in layer l and $l+1$, respectively, and where h is the output of unit i and w_{ij} is the weight connecting unit i and j. We can interpret Equation 4.22 as a measure of how important the output of unit i was for each of the units in the next layer. This measure is normalized to the other units of layer l and Equation 4.22 thus satisfies the conservation property from Equation 4.21. The same layer-wise propagation rule holds true for convolutional layers which can be regarded as a special case of a fully connected layer where weights are repeated multiple times (Section 4.4). Different activation functions can also be handled [158, 166, 173].
LRP thereby obtains relevance values that illustrate the importance of individual features for the entire input, given a data point of interest. Relevance values can be both positive or negative, indicating whether the presence of a feature has a positive or negative impact on the classification result. Because individual gradients tend to be noisy, multiple slightly different **LRP** rules have been derived [173–176] that absorb contradictory explanations [174] or allow giving weights to positive and negative explanations (reviewed in [176]). In this work the ϵ-**LRP** rule was used which adds a small constant to the denominator of the general **LRP** rule to absorb noise or contradicting gradients. The ϵ-**LRP** rule can be written as:

$$R_i^{(l)} = \sum_j \frac{h_i w_{ij}}{\epsilon + \sum_i h_i w_{ij}} R_j^{(l+1)}. \tag{4.23}$$

In practice it was shown that the basic **LRP** (and ϵ-**LRP**) propagation rule in Equation 4.22 can be expressed in terms of a modified gradient rule [166, 171, 172] when only ReLU activation functions are used, making LRP compatible with modern deep learning frameworks such as Tensorflow [177]. Ancona et al. [171] reformulate the **LRP** rule through partial derivatives with respect to the internal mappings H^l and compute the values using a propagation rule more similar to the backpropagation algorithm (see Kindermans et al. [172] or Ancona et al. [171] for the mathematical proof). Assuming top layer relevances $R^{(l+1)}$ corresponding to the nodes of $H^{(l+1)}$, the relevance for the nodes in $H^{(l)}$ can be computed as:

$$R^{(l)} = H^{(l)} \cdot \frac{\delta H^{(l+1)}}{\delta H^{(l)}} \cdot \frac{R^{(l+1)}}{H^{(l+1)}} \tag{4.24}$$

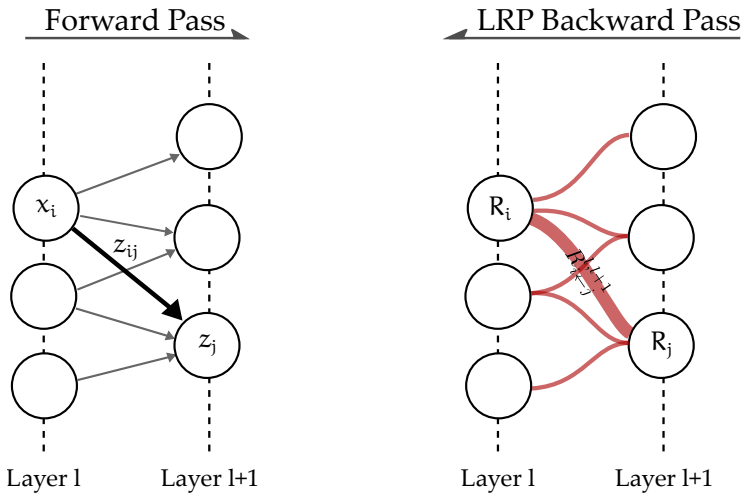


Figure 4.16: **The workflow of layer-wise relevance propagation (LRP).** It attempts to redistribute the total amount of relevance $R = f(x)$ to the units of the network. Network weights are indicated as edges and units (or neurons) correspond to nodes. Larger weights (w_{ij}, connecting $x_i^{(l)}$ and $x_j^{(l+1)}$) produce a higher flow of relevance during the LRP backward pass. (Figure reproduced from [166])

where $H^{(l)}$ denotes the intermediate representation learned by the GCN at layer l. Connecting **LRP** decomposition steps for consecutive layers yields:

$$R^{(Input)} = X \cdot \frac{\delta H^{(1)}}{\delta X} \cdot \dots \cdot \frac{\delta H^{(L)}}{\delta H^{(L-1)}} = X \cdot \frac{\delta f(X)}{\delta X} \tag{4.25}$$

where $H^{(0)} = X$ corresponds to the input data vector or matrix X and $R^{(Input)}$ to the vector or matrix of relevance values for each input feature.

4.7 SUMMARY

This chapter introduced some of the fundamental problems of **ML** as well as methods to solve them. We saw that non-linear models are able to learn more complex tasks but at the price of having more free parameters which makes them prone to overfitting. Neural networks are a class of non-linear models that transform data in layers to yield representations. Those representations are more and more capable of solving the classification or regression problem at hand. Convolutions allow introducing domain bias to neural networks. Graph convolutions achieve the same for data sets embedded in graph structures and thereby provide a powerful classification framework to combine relational and feature data. Finally, neural networks can be interpreted by exploiting their differential and hierarchical architectures.
The next chapter will introduce EMOGI, an interpretable method based on **GCNs** to predict cancer-related genes from multi-omics features and relational protein-protein interactions.

5

GRAPH CONVOLUTIONAL NETWORKS FOR PAN-CANCER DRIVER IDENTIFICATION

In Section 2.2, we have seen that the precise linking of genotype to phenotype is a hard problem and requires an almost complete understanding of gene regulation. And because cancer malignancies are evolutionary diseases of the genome, many if not all cellular processes can be hijacked and corrupted in order for cells to acquire a growth advantage and grow outside of their otherwise tightly regulated microenvironment. But Chapter 3 gave an overview of additional experimental approaches that provide shortcuts and approximations of cell states. **RNA-seq**, for instance, directly measures the amount of produced **mRNA** in a bulk of cells and can give hints on gene products involved in cancer-related processes without having to link complex genomic changes in non-coding regions, epigenetic readers/writers or other molecular mechanisms to them.
However, the battery of *"-omics"* data available nowadays poses significant challenges on the computational side of cancer genomics. The vast amount and heterogeneous nature of different *omics* require tailored machine learning algorithms. They have to be able to integrate data from different representations, generalize well in the face of scarce training sets but still process giga and terabytes of high-dimensional data.
This chapter will introduce an **E**xplainable **M**ulti-**O**mics **G**raph **I**ntegration (EMOGI) model that integrates **Single Nucleotide Variants**, **Copy Number Aberrations**, DNA methylation in promoter regions and gene expression data with **Protein-Protein-Interactions** in an explainable machine learning model. EMOGI is based on **GCNs** (introduced in Section 4.5) and classifies genes into cancer-related and cancer-unrelated genes. It is an extension of the **GCN** model introduced by Kipf & Welling [133]. The interpretation of classification decisions is done with **LRP**, introduced in Section 4.6.3, and allows to gain insights into the cancer types as well as data types important for the classification of a gene as cancer-related gene and further permits to attribute parts of the relevance to the **PPI** partners of a gene of interest.

5.1 THE EMOGI GRAPH CONVOLUTIONAL MODEL

A **ML** model that is able to reliably predict cancer genes has to incorporate different data types and representations. Mutation rates [29], copy number changes [20], epigenetic states [61] and the expression of genes [37] have proven essential data types to understand cancer processes in the past (introduced in Section 2.2.2). Furthermore, **Protein-Protein-Interactions** carry orthogonal, non-redundant information about protein complexes and pathways that are highly relevant for the detection of cancer genes [41, 44, 49]. The goal of this study is gaining knowledge about cancer processes based on the all of the data types mentioned above. To that end, interpretability is absolutely crucial. An interpretable model, however, requires that the data is integrated early and in one model jointly. That way, redundancies and re-

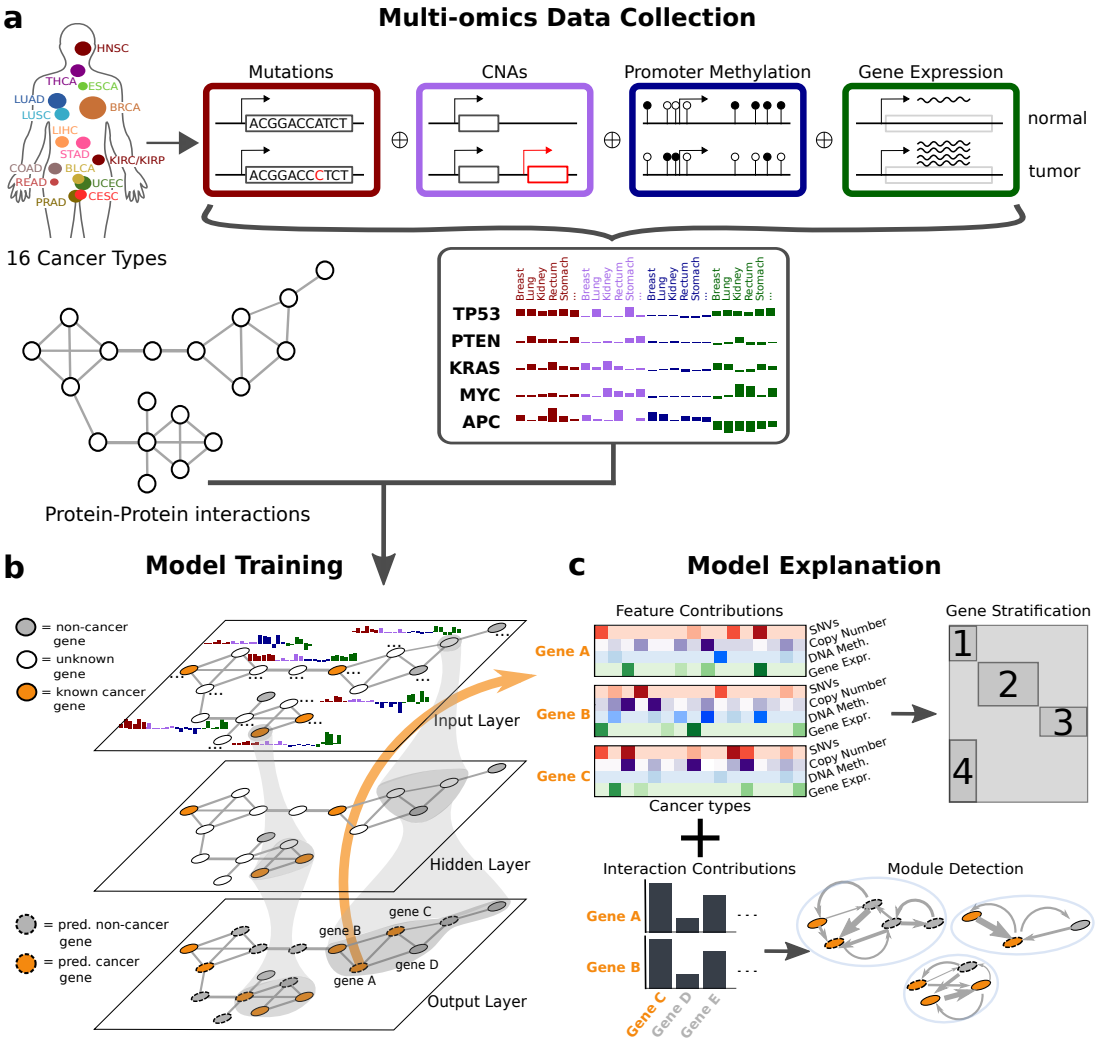


Figure 5.1: **EMOGI data collection, model training and explanation. a** Different Omics-levels from more than 10,000 patient biopsies (see Table A.1 for detailed numbers) were collected from TCGA [30]. Specifically, **SNVs**, **CNAs** (experimental approaches to detect mutations are discussed in Section 3.2), DNA methylation profiles (whose experimental background is described in Section 3.3) and gene expression data (experimental techniques described in Section 3.4) were collected. Each data type is individually processed according to standard procedures in the field to the point where a matrix of genes $\times$ cancer types is obtained for each of them. Those matrices are concatenated to form the feature matrix X which is used as input to EMOGI alongside a **PPI** network represented as a graph where nodes correspond to genes and edges denote interactions between their proteins in the cell. **b** A **GCN** is trained in a semi-supervised fashion on the partially labeled graph where nodes carry multi-omics feature vectors. **c** The **LRP** framework is used *a posteriori* to interpret EMOGI and to identify classes of cancer-related genes and modules that drive cancer progression and maintenance.

occurring patterns in the different data sets are discovered directly and can guide the classification. An alternative approach would be to first construct a **ML** model for each data type and join their learned data representation. This process is often referred to as late integration.
The molecular processes within a cell are highly complex and dynamic. The orthogonal data types are heterogeneous and measure different states as well as molecular processes. It is therefore very likely that any predictive modeling approach trying to predict cancer associations for genes has to incorporate non-linear relationships in order to make sensible predictions.

EMOGI combines somatic mutation information in the form of **SNVs** and **CNAs** with DNA methylation at promoter regions and gene expression data for different cancer types. Each data type is preprocessed to obtain a gene-sample matrix (the data collection and preprocessing steps are explained in Section 5.2). Figure 5.1 gives an overview of the EMOGI workflow. The data is preprocessed in a gene-centered manner to obtain a feature matrix X where each gene corresponds to a data point. X has a row per gene and the columns represent the different cancer types and omics levels. According to the early integration scheme, the feature matrices for each omics level are subjected to normalization and then concatenated to form X (this is explained in more detail in Section 5.6). This feature matrix and a **PPI** network are used together for training a **GCN** model that learns to distinguish cancer-related from cancer-unrelated genes. Proteins in the network are considered products of their genes and therefore, **PPIs** are treated as interactions between genes. EMOGI solves a binary classification problem using **Known Cancer Genes** from different expert-curated databases that serve as positive labels, as well as negative labels (non-cancer genes) that are obtained by recursively filtering all genes. Any gene that might have an association with cancer is removed from the set of all genes until a set of genes is reached that has no association with cancer with high probability (details in Section 5.2.6).
The labeled genes are used to compute the loss (details on how the loss is computed are given in Section 5.3), backpropagation (explained in Section 4.3.2) is used to compute the new weights iteratively and the ADAM optimizer [178] is used as **Gradient Descent** method. Training starts with small random weights drawn from a normal distribution [117]. At each iteration of the **GD** algorithm (called *epoch*), weight updates are computed from the whole batch of training data (X and the **PPI** network) and this procedure is iterated until convergence (usually around -5,000-10,000 epochs). Regularization such as dropout and norm penalties (explained in Section 5.4) are used to avoid overfitting and help generalization.
After training of EMOGI, **LRP** [151, 158] is used to map the classification output back to the input for any gene of interest. The interpretation gives most important features and interaction partners for any gene of interest.

5.2 DATA COLLECTION & PROCESSING

Figure 5.1a shows how the EMOGI method combines different publicly available data sets and integrates them to form a feature matrix X, a graph G and labels y that serve as input for the **GCN** algorithm. This section will explain in more detail how the

data was collected and processed to be most informative to a computational method. Generally, relative values that compare a quantity of a feature in tumor compared to a normal cell from the same cell type are more informative than absolute numbers from cancer samples alone to predict associations with cancer diseases. For somatic **SNVs** and **CNAs**, the difference from a normal genome is given by definition. For both, gene expression and DNA methylation, however, this is not the case. Both of these data types have sharp, tissue-specific patterns as introduced in Section 2.1.2 (see Figure A.2 for UMAP embeddings of all *omics* levels into a two-dimensional space) and are expected to be informative only as relative measurements that compare tumor and normal tissue samples. Fortunately, **TCGA** also contains normal samples from adjacent tissues of the same type for several cancer types (see Table A.1 for the number of samples per *omics* level) and makes it possible to construct relative measurements between tumor and normal tissue.

In total, **SNVs**, **CNAs**, DNA methylation and gene expression data of more than $8,000$ samples were collected from **TCGA**, covering 16 different cancer types. Table A.1 lists all 33 available cancer types from **TCGA** alongside the corresponding number of samples for each of them, and whether they could be used for EMOGI. The analysis was limited to those cancer types for which DNA methylation information in tumor and normal tissue was available and for which already pre-processed gene expression data from Wang et al. [55] existed. For the other cancer types, either DNA methylation or gene expression in the adjacent normal tissue was lacking. Unfortunately, all leukemias (blood cancers) had to be removed from this study because normal blood samples can no longer be taken from patients with blood cancers. Theoretically, it is possible to incorporate those cancer types as well but it would require samples from healthy donors that serve as normal control. Such a normalization using different donors, however, can result in significant biological and experimental biases because different patients may have different expression levels (biological variation) and experimental molecular methods (**RNA-seq** and methylation assays) have been reported to suffer from batch effects [179, 180].

All four *omics* data sets for the same 16 cancer types were pre-processed individually and finally concatenated to form a $N \times (4 * 16)$ matrix where N corresponds to the number of genes (see Figure 5.1). **PPI** data was collected from different sources. Genes that are not present in the largest connected component of the **PPI** network were removed from the feature matrix. To ensure that the scale of the individual feature matrices is roughly similar, min-max normalization was applied prior to the concatenation, although neural networks are expected to deal with differently scaled features by adapting the weights accordingly [133]. The resulting feature matrix exhibits clear patterns of correlations across genes, which are displayed in Figure 5.2. Each *omics* level is more correlated with itself than with other data types, justifying the use of normalization to scale the data. Furthermore, closely related cancer types, such as colon and rectal cancers exhibit higher correlations across all *omics* levels. And finally, a medium high anti-correlation between gene expression and DNA methylation data is biologically meaningful because of the silencing effect of DNA promoter methylation (further explained in Section 2.1.2). **CNAs** are less correlated due to the sparsity of **CNAs** in the samples.

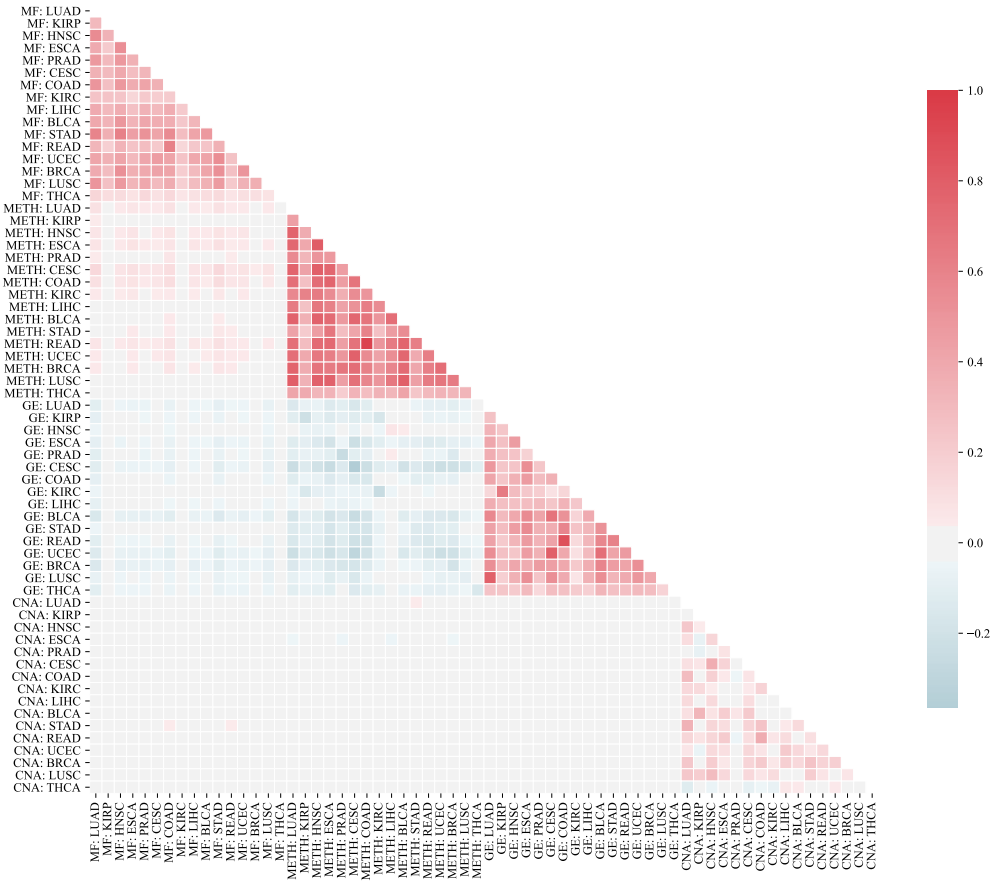


Figure 5.2: **Correlations of features with each other over cancer and data types.** After data collection of mutation rates, DNA methylation at promoters and gene expression the Pearson correlation between all of them are depicted. The color indicates correlation versus anti-correlation, and the deeper the color, the higher the correlation observed (legend on the right). Because correlations are symmetric, only the lower triangle is depicted.

5.2.1 *Single Nucleotide Variants*

TCGA releases somatic mutation data in a Mutation Annotation Format (MAF). Those files contain all detected somatic mutations in a cancer type. Each row represents a somatic mutation along the patient it occurred in. Calling mutations from short-read **DNA** sequencing is done by MuTect2 [181, 182]. The tool calls (or predicts) a variant from the supporting reads using a statistical significance test. Several tools for variant calling have been proposed over the years [28, 79, 181–184] and each has its own advantages. Furthermore, variant callers usually classify mutations into missense mutations that change the protein **AA** sequence, nonsense mutations that introduce a premature stop codon and silent mutations that do not change the resulting protein sequence. Silent mutations were removed in the EMOGI data preprocessing to reduce the number of random passenger mutations and concentrate on mutations that have an effect on the **mRNA** or the protein.
There are a significant number of mutations occurring randomly in the genome [25,

185]. Thus, longer genes are expected to accumulate more mutations, no matter if they associate with cancer diseases or not. Famously, the very long *TTN* gene—an essential component of sarcomeres that is important for muscle tissues—is sometimes used as a marker gene for methods that do not correctly account for gene length [40]. For the EMOGI preprocessing, the mutation rate (or mutation frequency) mr_{gp} for a gene g in a patient p was normalized according to:

$$\tilde{mr}_{gp} = \frac{mr_{gp}}{1 + bpmr_p * l_g} \tag{5.1}$$

where l_g denotes gene length and $bpmr_p$ denotes the patient-specific mutation rate per nucleotide, i.e. the overall probability that any base in the genome is mutated. Furthermore, some samples in **TCGA** were reported to be ultra-mutated due to extreme genomic instability. Those were removed using a list of known ultra-mutated samples that was extracted from Synapse (syn1729383). The preprocessing for **SNVs** roughly follows the preprocessing pipeline from HotNet2 [41]. No differences were made between loss-of-function mutations, missense mutations or gain-of-function mutations. This is because when a gene is more often mutated than expected, this already implies a selective growth advantage to the cell. The directionality of that genomic change (i.e. if it disables a gene or gives it new interaction partners or other new function) is not crucial to infer an association and hence not distinguished in EMOGI.

5.2.2 *Copy Number Aberrations*

DNA sequencing can also be used to detect **CNAs** of genes (see Section 3.2 and Figure 3.1 for details). **CNAs** arise through larger insertions or deletions in the genome and frequently occur in tumors because the DNA repair mechanisms are increasingly damaged. Amplified genes have more copies present in the genome while deleted genes have only one or no copies left from which **RNA** and finally proteins can be produced. The tool GISTIC2 [84] was used in this work to estimate the target genes of **CNAs**. The tool operates by identifying copy number profiles from mapping algorithms and then deconstructing them into individual somatic **CNA** events using a sophisticated background rate of **CNAs**. GISTIC2 also infers target genes from the identified **CNAs** events. Copy number gains and losses, corresponding to amplified or deleted genes were identified in the TCGA cohort for the 16 cancer types using GISTIC2. The results from GISTIC2 were downloaded via firehose from [1]. As for the **SNVs**, ultra-mutated samples from syn1729383 were removed, and the copy number rate of a certain gene was defined as the number of times a gene was amplified or deleted in a specific cohort. No difference was made between gains or losses in this context because EMOGI predicts cancer-related genes and not **TSGs** or oncogenes specifically. In theory, one would expect **TSGs** to be more often deleted while oncogenes, such as the famous oncogene *MYC*, are often amplified in cancers [20].

1 https://gdac.broadinstitute.org

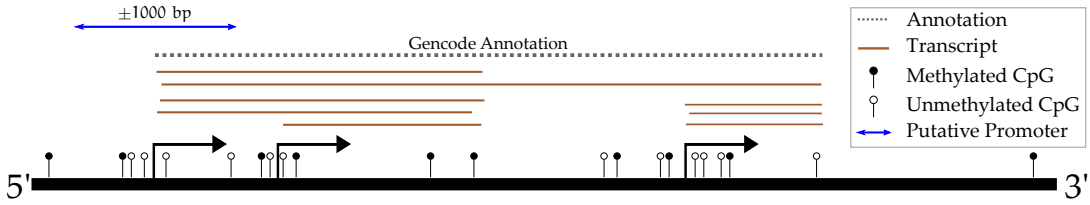


Figure 5.3: **Preprocessing of DNA methylation for a single gene.** Depicted is a gene with three alternative **TSSs**. To extract a meaningful DNA methylation of the promoter region for that gene, the Gencode annotation [186] was used to extract the coordinates of the **TSS** of the most 5′ transcript. Next, the promoter region was defined as the region ±1000 **Base Pair (bp)** around the **TSS** and the average β value in that region was computed.

5.2.3 *DNA Methylation in Promoter Regions*

DNA methylation data from the 450k Illumina bead array platform (described in Section 3.3) was collected from **TCGA**. The platform measures the very same 450,000 **CpG** sites in the genome for each sample. **CpGs** sites are selected by the manufacturer such that promoters, gene bodies and regulatory regions are sufficiently covered [87, 88]. The promoter region of a gene was defined as the ± 1000 base pair region around the **Transcription Start Site (TSS)** of that gene. Due to alternative splicing and slightly differing annotations, different transcripts exist for the same gene. For the EMOGI preprocessing, the transcript that is located farthest in the 5′ direction was used to select the **TSS** and corresponding promoter window (see Figure 5.3). Gencode (V 28) [186] was used as annotation for the genes. Within a promoter region of a gene, the average β value across all **CpG** sites was used to determine the DNA methylation status in the promoter. This value ranges between 0 and 1 and describes the portion of cells in the sample that were methylated. DNA methylation experiments are — similar to **RNA-seq** experiments — very susceptible to batch effects. Those effects arise because different sequencing facilities and hospitals introduce small unwanted technical variation that cannot be avoided and often, most of the variability in the data is explained by the laboratory or hospital taking/sequencing the sample rather than a biological difference. ComBat [187] is a tool that was proposed to deal with batch effects. It employs an empirical Bayes latent variable model to reduce batch effects generally in **NGS** data. For the methylation data, the plate number of the samples was used as batch variable the model[2]. ComBat was used independently for each cancer type.

For each gene i, the measure of differential DNA methylation at its promoter in cancer type c is defined as dm_i^c. This value describes the difference in methylation signals between cancer sample β_i^t, and matched normal sample β_i^n, averaged across all samples S_c available for cancer type c. It can be written as:

$$dm_i^c = \frac{1}{|S_c|} \sum_{s \in S_c} \left(\beta_i^t - \beta_i^n \right). \tag{5.2}$$

2 https://docs.gdc.cancer.gov/Encyclopedia/pages/TCGA_Barcode/

dm_i^c was computed for all genes across all 16 cancer types and aggregated in a matrix of genes and cancer types.

5.2.4 *Gene Expression*

To measure the expression level of each gene in each sample, data from Wang et al. [55] was used. In the study, **RNA-seq** data (introduced in Section 3.4) of both tumor and control samples from **TCGA** are combined with gene expression data from the GTEx consortium. The authors have realigned the reads to the reference genome, quantile-normalized the data and batch-corrected it using ComBat [187]. The data set was reported to be better suited for differential expression analysis because the scales of normal and tumor samples were more similar compared to the original released expression data from **TCGA** [55].
For each gene, differential expression was computed as log2 fold change between expression in cancer versus a matched normal sample. If a normal sample from the same patient (and the same tissue) was available, this sample was used to compute the fold changes. In cases where that was not available, the median gene expression value from all other normal samples was used for normalization. To verify that samples were correctly normalized, MA plots were computed and depicted in Figure A.1 for all 16 cancer types. In those, when the majority of genes is not located on the blue line (that indicates no change), tumor and normal samples are not on the same scale. This, however, is not the case for the data set from Wang et al. [55].
Similar to the DNA methylation, all samples from the same cancer type were averaged to obtain a matrix of genes and cancer types.

5.2.5 *Protein-Protein Interaction Networks*

Section 3.5 has introduced how **PPIs** can be experimentally measured. Each experiment, however, only yields small parts of the puzzle that the interactome poses. Therefore, multiple databases have started to fuse **PPI** information in human cells from different large-scale studies [94, 95, 188–190]. These databases usually contain scores that quantify the certainty of interactions between proteins and compare evidence from literature mining, **Y2H**, **TAP-MS** and other approaches. In total, these networks have collected between $100,000$ (IRefIndex) and up to 5 million (STRING-db) interactions. All of the networks have quite different characteristics and can yield very different results in the recovery of disease genes [190].
For EMOGI, five different networks — each of which incorporates hundreds of studies — were evaluated, namely STRING-db [95], IRefIndex [188], Multinet [189], PC-Net [190] and **Consensus Path DB (CPDB)** [94]. Two different versions of IRefIndex (one from 2015, one from 2019) were compared to gain knowledge about the progress and changes of **PPI** networks over time, yielding 6 different networks in total.
Each of the networks was preprocessed slightly differently. For **CPDB** and STRING-db, only high confidence interactions were considered, using a cutoff of 0.5 for CPDB and a cutoff of 0.85 for STRING-db, respectively.
Multinet and the old version of IRefIndex were collected from the Hotnet2 github

repository[3]. For the most recent version of IRefIndex, only binary (involving only two proteins) and human interactions were considered. Finally, PCNet was not pre-processed further. For all networks, edges with weights above the respective threshold were selected while the others were discarded. This resulted in unweighted and undirected networks, similar to the original application of **GCNs** to citation networks.

5.2.6 *Positive and Negative Examples*

In Section 4.5.3, we saw that **GCNs** are a semi-supervised machine learning method. To improve the randomly initialized weights, they evaluate a loss function on labeled data points. While these only represent a fraction of the nodes in the graph, it is crucial that the labels are accurate.
Much work was dedicated to developing a catalogue of cancer driver genes in the past [31, 191–199] and has led to databases such as the COSMIC cancer gene census (CGC) [195], OncoKB or the **Network of Cancer Genes (NCG)**. To compile a set of known cancer genes that serve as labels for the positive class (cancer-related genes), 711 **Known Cancer Genes (KCGs)** from the **NCG** were collected. The **NCG** comprises a manually curated list of cancer genes, highly overlapping with the COSMIC CGC (see Figure A.7 for overlaps between different sets of **KCGs**). The vast majority of known cancer genes have been annotated as such because they were mutated more often than expected by chance in large cohorts of patients [19]. This is a problem for a multi-omics setup where the goal is to find cancer genes based on multiple sources of evidence. To obtain a more diverse set of known cancer genes without losing the accuracy of manually curated data, the known cancer genes from the **NCG** were extended by gene sets from DigSEE [199]. This database represents a set of high-confidence cancer genes mined from PubMed abstracts and additionally categorizes them by the type of evidence that supports it. This way, additional 43 cancer genes that have observed changes in DNA methylation and 137 cancer genes with altered gene expression patterns were included in the list of positive labels.
Negative labels (e.g. cancer-unrelated genes) were harder to obtain as there are no databases that collect non-cancer genes and hence, those had to be collected through a filtering approach. The idea behind the filtering is to consecutively remove all genes that could have an association with cancer diseases until only a sufficiently small set of genes remains. Firstly, all genes that are part of the positive labels were removed. Next, all genes contained in the OMIM database [200] were removed. This excludes all known disease genes for any disease. While this step removes many genes that have no association with cancer, it helps to make sure that prominent disease genes no longer are present in the set of non-cancer genes. Next, all genes that belong to known cancer pathways from KEGG [201] were removed as well as genes that were reported to be significantly more mutated than expected in large-scale cancer screens (using a set of significantly mutated genes from the COSMIC database). Finally, genes predicted to be involved with cancer by MutSigdb [202], as well as genes whose expression was found to be correlated to the expression of cancer genes [203] were removed. For the application with EMOGI only genes present in the **PPI** graph were

3 https://github.com/raphael-group/hotnet2

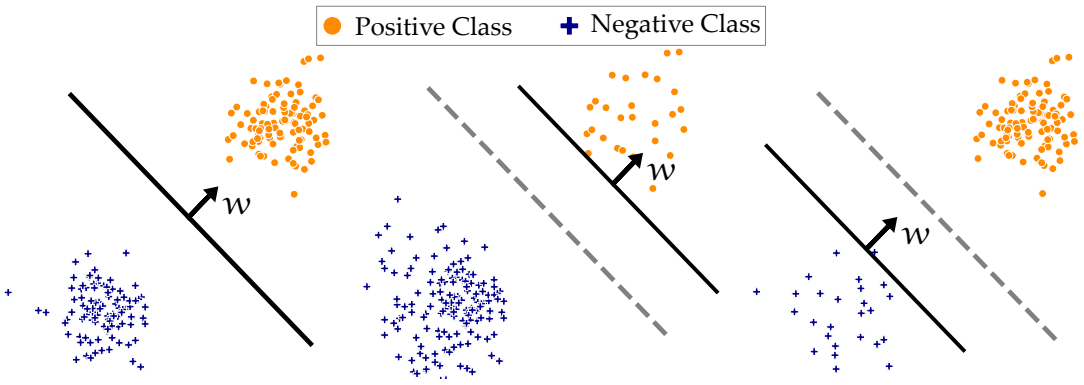


Figure 5.4: **Different number of positive and negative examples shift the decision boundary.** Depicted are three settings of a binary classification: (left) Both classes contain roughly the same number of data points and therefore, the decision boundary is located exactly between the two classes and has a maximum distance to points from both classes. (middle) The positive class is underrepresented, resulting in a shift of the decision boundary towards the positive class. (right) The negative class is under-represented and the decision boundary is shifted towards the negative class. The distribution of the data points did not change between the three scenarios. The gray dotted line depicts the original balanced decision boundary.

used. The final set of labels thus depends on the chosen **PPI** network. For the **CPDB PPI** network, for instance, it comprises 796 positively and 2187 negatively labeled genes.

5.3 CLASS IMBALANCE

In the previous section, we saw that the number of labels for the positive class (cancer-related genes) is much smaller than the negative class of non-cancer genes. This can pose a problem when computing the loss because the class with more labeled data points (the majority class) might dominate the classification (see Figure 5.4 for a visualization of how the decision boundary shifts with class imbalance). The extreme case would be a classifier that learns to only predict the majority class for all data points because the minority class does not increase the loss sufficiently. Such a case represents a situation where the optimization converges to a local minimum that might give small losses and high accuracy but is not desired.

Solutions to deal with class imbalance typically include oversampling of the minority class, subsampling of the majority class or directly modifying the loss function to give higher weight to the minority class. Oversampling means using data points from the minority class multiple times per epoch and is not straightforward to implement for **GCNs** because the graph structure would require that the oversampled genes are embedded in the graph somehow. Subsampling the majority class, on the other hand, is undesired because it would significantly reduce the number of labeled genes and add another stochastic element to the training process.

Thus, direct modification of the loss function was used to scale the loss for the minority class by a certain factor. This factor is considered a **Hyper-Parameter (HP)** to the

model and optimized alongside other **HPs** as discussed in Section 5.7.
For EMOGI, a binary cross-entropy loss was used for training which is common practice for modern **Deep Neural Networks (DNNs)** [99]. To give more weight to the positive class, this loss function can be modified as:

$$\mathcal{L} = -\big(w_{pos} y \log(\hat{y}) + (1-y)\log(1-\hat{y})\big) \tag{5.3}$$

where w_{pos} denotes the scaling factor for the positively labeled points and $\hat{y} = f(x)$ denotes the output from the **GCN**.
This way, the original amount of training data remains unchanged and the additional weight given to the minority class can be optimized along with other **HPs**.

5.4 GCN REGULARIZATION

In total, 2187 labeled genes were obtained for the **CPDB PPI** network which contains $13,627$ genes, corresponding to a labeling rate of 15%. At the same time, the EMOGI model has roughly $10,000$ trainable weights with the exact number very much depending on the model architecture that is selected during **HP** optimization (explained in Section 5.7). This means that the number of trainable weights is significantly higher than the number of data points (termed $p \gg N$ problem and closely linked to the curse of dimensionality and introduced in Section 4.2.1). A successful model therefore has to be rigorously regularized to generalize well. Regularization of **ML** algorithms generally aims to reduce model complexity and therefore variance through the increase of the model bias (see Section 4.2 or the book by Goodfellow et al. [99]).
Regularization of neural networks is currently an active topic of research and many different techniques have been proposed in the recent years. Among the most successful ones are norm penalties and dropout [99]. While the former is a general **ML** regularization method also applicable to linear regression and other methods, the latter is a method specifically designed for neural networks. EMOGI makes use of both of them and effectively increases bias, reduces variance and improves its generalization to data beyond the training set.

5.4.1 *Norm Penalties*

Norm penalties are a common technique used in **ML** and were shown to be effective also for **ANNs** [99, 108, 204]. They are often referred to as **weight decay** or **shrinkage** in different subfields. The idea behind norm penalties is the addition of a regularization term to the loss function that encourages small values for the parameters unless the data suggests otherwise [104]. Generally, a norm penalty for an objective function $\mathcal{L}$ can be written as:

$$\tilde{\mathcal{L}}(\Theta, X) = \mathcal{L}(\Theta, X) + \alpha\Omega(\Theta) \tag{5.4}$$

where Θ denotes the model parameters, Ω denotes a norm penalty function and α corresponds to a weighting of the original objective ($\mathcal{L}$) and the penalty (Ω). Different norm penalties can be used, with the most common ones being the L^1 norm (known

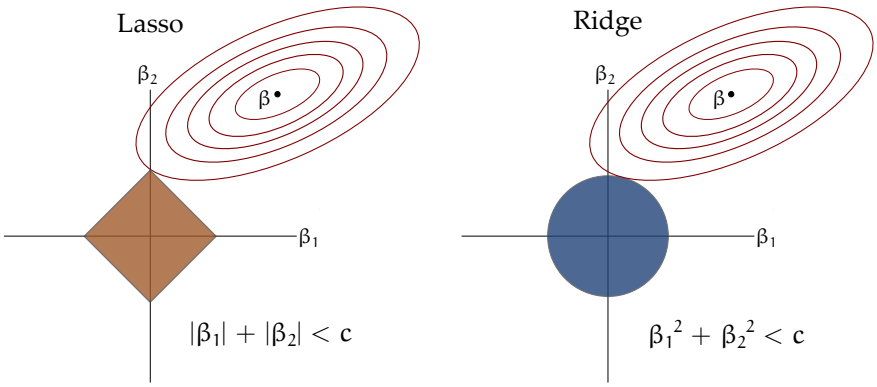


Figure 5.5: **LASSO (L^1) and Ridge (L^2) norm penalties.** The figure depicts a fictive two-dimensional optimization problem whose optimal solution is denoted by β. The red circles indicate the distance from β. The diamond (for LASSO) and the circle (for Ridge) represent the constrains to the parameters β_1 and β_2. Lagrangian constrained optimization attempts to find a point that fulfills both optimization criteria by selecting the point within the diamond or circle that it is closest to the optimal solution β. For LASSO, this would result in β_2 to be 0 which is desired in linear problems.

as LASSO [205]) and the L^2 norm (known as Ridge regularization [206]). Norm penalties are a specialized version of Lagrangian constrained optimization where the addition of a second objective (like setting most parameters to 0) is in conflict with the original objective (fitting the data as well as possible) and a compromise has to be found. Figure 5.5 visualizes this conflict for a two-dimensional problem for the LASSO and Ridge norm penalties.
In linear problems, LASSO (L^1 norm) is often preferred because of the diamond shape that shrinks most model parameters to exactly 0, while Ridge regularization (L^2 norm) shrinks to a value close to 0. Therefore, LASSO uses the least amount of parameters possible while still obtaining a good fit of the data, making the model more interpretable (Figure 5.5) [104, 106].
In neural networks, the weights (especially in higher layers) do no longer have a direct meaning and hence, it is not preferred to have them set to exactly 0. Nonetheless, the penalty enforces a competition between the weights because the overall size of all weights is bound. Therefore, features that do not contribute to the data fitting objective will have smaller weights associated with them while important patterns are associated with higher weights. Hence, norm penalties help to reduce variance and increase bias of the model by preferring smaller values to larger ones.
For this work, L^2 norm penalties were used to encourage smaller weights for all layers.

5.4.2 *Dropout*

Ensemble methods such as random forests or boosting build several weak learners and average their outputs to obtain a more robust estimate of the prediction. Interestingly, an ensemble of models can always increase model performance or at least

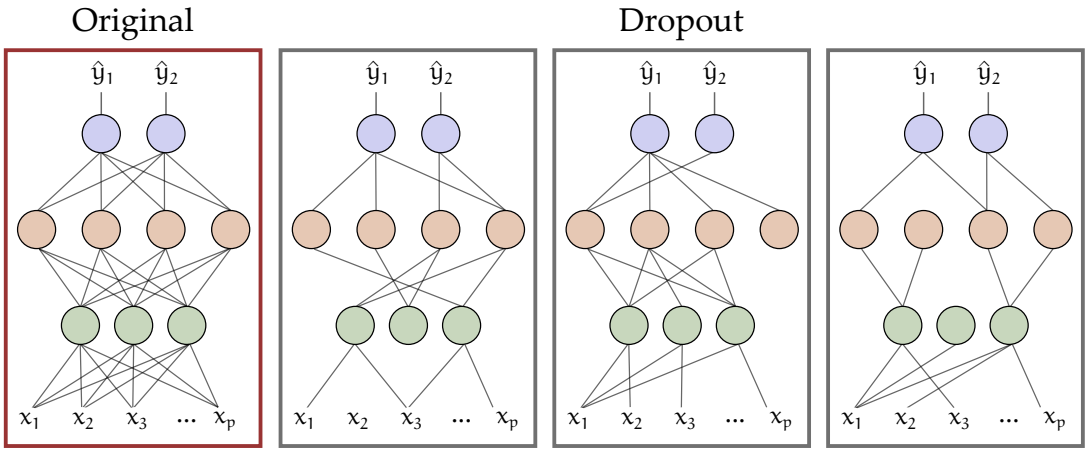


Figure 5.6: **Dropout regularization through dropping random connections in a neural network.**The network on the top depicts a fully connected neural network with two hidden layers. During the training process, a fraction of connections is randomly chosen and set to 0 and this process is referred to as dropout (depicted on the bottom). This makes the network learn important patterns multiple times. Another view of dropout is that the sparse networks (depicted on the right) all correspond to different models. The final fully connected model corresponds to an ensemble classifier [99].

result in the original performance when the errors of the individual models are perfectly correlated [99, 108]. This makes ensembles a powerful method to use in a broad variety of applications. Unfortunately, neural networks are already computationally expensive to train, making larger ensembles of neural networks often infeasible.

Dropout is an approximation of ensemble learning without having to train multiple models [207]. Instead, the iterative training procedure is used to train a slightly different model at each epoch. Randomly, a fraction of the connections in the neural network are set to 0, training an exponential number of different thinned networks in parallel (visualized in Figure 5.6).

Intuitively, dropout prevents overfitting since individual units cannot gain very high importance in the network because the risk that these units are dropped at some point is very high. Dropout encourages learning patterns and concepts redundantly, making the resulting model more robust.

During test time, dropout is no longer applied which gives the effect of averaging across all the thinned networks, similar to the majority vote taken by other ensemble models [207].

5.5 EXTENSION OF GRAPH CONVOLUTIONS TO FEATURE TENSORS

EMOGI was adapted to perform gene classification for a specific cancer type at the level of individual patients, without averaging features across the samples of a given cancer type, but using the *omics* values of the single patient samples for training. In order to do so, the graph convolution operation defined in [133] (and described in Section 4.5.3 and Equation 4.19) was adapted to work with a feature tensor of rank 3. In detail, a feature tensor $F \in \mathbb{R}^{N \times S \times C}$ was constructed, harboring one "dimension"

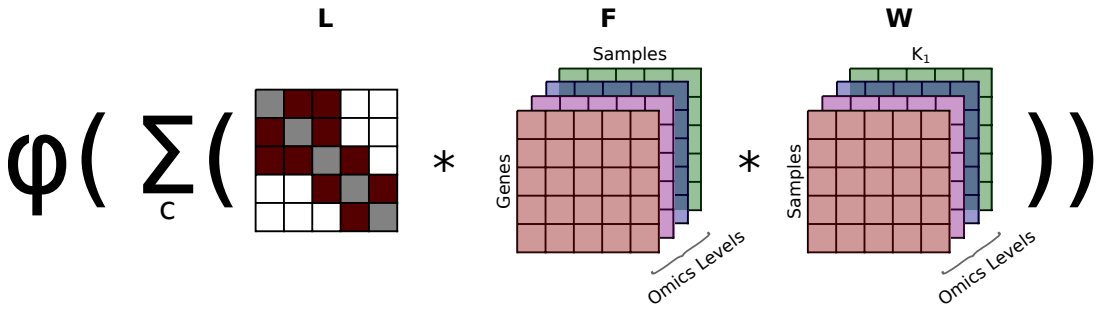


Figure 5.7: **Graph convolution operation for rank 3 tensors.** The adapted graph convolution operation takes one "slide" (corresponding to a single *omics* level) of the feature tensor F and the weight tensor W along with the graph Laplacian L to perform a graph convolution (slides are indicated by different colors). This produces 4 activation maps (for **SNVs** in red, **CNAs** in purple, DNA methylation in blue and gene expression in green) that are summed together after the convolution. This procedure is analogous to how rbg channels of color images are convolved in CNN models for image processing [120].

for the genes (N), one for the patient samples (S) and one for the *omics* levels (C), as summarized in Fig. 5.7. The different *omics* levels were treated similarly to the rgb channels of a color image processed by a **CNN** model. This led to the following formulation of the convolution operation for the first model layer, leading to the activation map H^1:

$$H^1 = \sigma\Big(\sum_{c=1}^{C} L * F_{::c} * W^{(0)}_{::c}\Big) \tag{5.5}$$

where $W \in \mathbb{R}^{S \times H \times C}$ denotes the weight matrix to be learned (similarly to Equation 4.19 in Section 4.5.3), H is the dimension (number of units) of layer 1 and L the graph Laplacian. The activations of the separate *omics* levels $c \in 1, ..., C$ are added after convolving every *omics* level separately with the graph Laplacian L and the weight matrix W. This is analogous to what is usually done with the rgb channels of a color image in a **CNN**. The extended graph convolution is depicted in Fig. 5.7. This produces an activation map ($H^{(1)}$) that has the same dimensionality as the activation map of the first layer defined in the original model (with a two-dimensional feature matrix as input, see Section 4.5.3). The successive layers are hence not affected by the use of a rank 3 tensor, and the previously defined graph convolution operation (Equation 4.19) can be used for all layers after the first one.

5.6 MODEL TRAINING

EMOGI is trained using a feature matrix X (derived using multi-omics data and explained in Section 5.2), a **PPI** network encoded as adjacency matrix A (explained in Section 5.2.5) and sets of known cancer and non-cancer genes (derived in Section 5.2.6). X is derived from four individual feature matrices for the individual omics levels (**SNVs**, **CNAs**, differential DNA methylation in promoter regions and differen-

tial gene expression). The four matrices intrinsically have very different scales. Therefore, min-max normalization was applied to force the values to have a similar scale. Min-max normalization rescales a data vector using the following equation:

$$x_{norm} = \frac{X - min(X)}{max(X) - min(X)}. \tag{5.6}$$

The normalization is required because some data types have a specific scale, such as DNA methylation in promoters which ranges between -1 and 1 while gene expression can be positive and negative but without a scale, and **SNVs** and **CNAs** information is only positive without a defined scale. After normalization, the four individual omics matrices are concatenated (depicted in Figure 5.1a) to form a feature vector of length $16 \cdot 4 = 64$ for each gene under consideration. The impact of the normalization is depicted in Figure A.3.
Conversely, unmeasured values for genes in the network (missing DNA methylation in the promoter of the respective gene, for instance) were set to 0. Hence, the final feature matrix X contains exactly those genes that are present in the **PPI** network.
The labeled genes are split into a training and test set (visualized in Figure 5.8), such that the ratio between positives and negatives is maintained in both sets. The training set contains 75% of the labeled genes while the test set contains the remaining 25% and is not used during the training or **HP** optimization.
Two neural networks might achieve the same high performance but can in theory classify data points based on different criteria. To gain knowledge about new putative cancer genes, it is not only important to interpret the features leading to the classification of a gene but also to know how certain the model is about it. Ensemble classifiers (introduced in Section 4.1.3) allow for an estimate of uncertainty through a potential disagreement between the weak learners. Similarly, if the weak learners are interpretable, estimates about the certainty of that interpretation can be made. To profit from that property, multiple EMOGI models were trained independently of one another with slightly different training and validation sets. The test set was not used at this stage for evaluation. The training sets for each of the models were computed using 10-fold **Cross-Validation** (**CV**) (explained in Section 4.2 and visualized in Figure 5.8). That is, the training set is once again split into 10 different sets to train 10 EMOGI models. Each of the models is trained on $\frac{9}{10}$ of the training data while the remaining 10% are left out to validate successful training. In the end, roughly 60% of the total labeled data was used for training each model. For model inference, an average vote of the 10 different models is taken. This way, the output probability of EMOGI actually reflects 10 different models that were trained with slightly less data. By querying each of the models separately for interpretation, estimates of the certainty of the interpretation can be obtained. Chapter 6 will show that the certainty of the interpretation actually correlates with the final output probability with higher output probabilities also giving more robust feature interpretations.
Each of the **GCN** models receives an adjacency matrix, multi-omics features and labels and computes several layers of graph convolutions using Equation 4.19. **ReLU** functions are exclusively used as non-linear activation functions. The last layer performs a graph convolution with a single filter and a sigmoidal non-linearity. Therefore, the output $f(X) = \hat{y}$ of an EMOGI model corresponds to a vector $\hat{y} \in \mathbb{R}^N$ in

which each entry denotes the probability of a gene to associate with cancer diseases. In each epoch, the weighted cross-entropy loss derived in Section 5.3 is computed and the backpropagation algorithm is used to compute derivatives of the weights. The ADAM optimizer [178] is used as an advanced form of stochastic **GD** and EMOGI models are trained for a fixed number of epochs.
The model architecture (number of filters per layer and number of graph convolutional layers), the learning rate η and regularization parameters are—among others - **HPs** to the EMOGI model. The next section discusses the optimization of **HPs** and reasonable ranges of parameters to expect.

5.7 HYPER-PARAMETER OPTIMIZATION

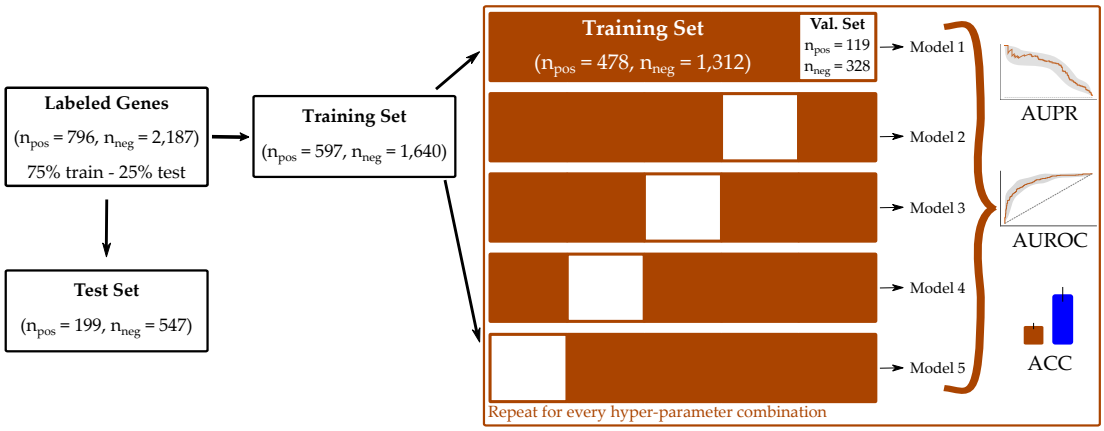


Figure 5.8: **Data splitting, CV and HP optimization.** The labeled genes are split into training and test sets, preserving ratios of positive and negative labels. The training set is further split into training and validation sets.

A key challenge to successful model training is the high number of parameters that can have crucial effects on the training procedure but are not trainable. A parameter that cannot be trained but is used to parametrize the model is called **Hyper-Parameter** (**HP**). Such parameters most often reflect design decisions in the modeling process such as weighting regularization against the objective function, specifying the architecture of the model or determining the length of the training procedure.
Table 5.1 lists the different **HPs** of EMOGI as well as reasonable parameter ranges that are used in the literature.
Unfortunately, there is no guarantee that the parameter ranges listed there actually correspond to good choices for the different **HPs** in the particular setting of predicting cancer-related genes. Therefore, **CV** can be used to find optimal or near-optimal combinations of parameters in combination with a grid search.
Optimizing **HPs** of **ML** methods is a time-consuming problem because each combination of **HPs** can yield unique results and there are no guarantees regarding continuity of the performance with respect to changes in the **HPs**. Consequently, all combinations have to be tried out in a brute-force approach. In the previous section, we saw that reasonable ranges for most **HPs** can be guessed from other applications even

Name	Description	Parameter Ranges
Number of Layers (L)	The number of graph convolutional layers used for training EMOGI.	2–3 layers were reported sufficient [133]. More layers can actually be harmful [142].
Number of Units per Layer	The number of units/filters per graph convolutional layer.	No prior knowledge.
Number of Epochs	The number of epochs/length of the training	Kipf & Welling use 200 epochs maximum with early stopping in place. Metrics per epoch can help assess if training time was sufficient.
Dropout Rate	Describes how many percent of the units are dropped during training time (Section 5.4.2)	Dropout rates of $0.5 - 0.8$ are recommended throughout the literature .[99, 207]
Weight Decay (λ)	Denotes the importance of weight decay/norm penalty regularization compared to the original objective (0 = no regularization, 1 = no data fitting, Section 5.4.1)	$1e^{-4}$ and $1e^{-5}$ were used in [133] but no other prior knowledge.
Learning Rate (η)	Describes how big the steps are per weight update.	The original publication used 0.01 but this parameter is tightly coupled with the number of epochs and weight decay. Typical values range from $0.1 - 0.0001$ [208] but also depend on the optimizer used.
Support (K)	Local support of the graph convolution (Section 4.5.2). $K = 0$ corresponds to a standard neural network, $K = 1$ corresponds to the **GCN** from [133] and higher values of K use Chebychev polynomials from [128].	Experiments with $K = 1$ and $K = 2$ were done but interpretability is further complicated with higher support. Therefore, $K = 1$ was used throughout this work and larger network structures were accounted for through multiple layers of graph convolutions.
Cancer Gene Weight	The factor by which **KCGs** are multiplied to alleviate the effect of imbalanced classes (see Section 5.3 for details)	No prior knowledge but ranges between 1 and 100 can be assumed reasonable and small changes are unlikely to change the overall objective drastically. Strongly interdependent with the learning rate η.

Table 5.1: **Hyper-Parameters of the EMOGI model.** Several hyper parameters influence the performance of EMOGI and good values are hard to estimate. This table depicts sensible default values and ranges to try from previous studies and helps to guide hyper-parameter search.

though there is no guarantee that a different combination of **HPs** gives much better results in practice.
CV can be used to estimate reasonable **HPs** for **ML** models and in particular for neural networks. By splitting the training set further into training and validation sets (the orange box in Figure 5.8), each of the k models in the **CV** can be trained independently. To test the behavior of different combinations of **HPs** empirically, one can use the validation set to assess the generalization error of a model. In practice, there are often data points that are harder to classify than others. Hence, computing k models with the same **HPs** results in a less biased estimate of how well the final model will generalize. To test which combination of **HPs** will most likely perform best on the test set a grid search will sequentially train k models for each possible combination and assess metrics, such as the **Area under the Precision-Recall Curve (AUPRC)** or accuracy (described in Section 4.2).
In Chapter 6, we will use the grid search in combination with **CV** to find reasonable **HPs** for EMOGI.

5.8 EXPLAINING EMOGI PREDICTIONS

To find multi-omics features of a gene and interaction partners of their proteins that drive the classification, **LRP** [158] was used (introduced in Section 4.6.3). This gradient-based, prediction-level interpretability method for neural networks was chosen for this work because it does not make assumptions about locality in the input, such as many perturbation-based methods. Occlusion, for instance, marks patches of an image which requires the assumption that neighboring pixels are interdependent (see Figure 4.15). Local linear models produce a faithful classifier in the surroundings of the data point of interest but it is unclear if its explanations really correspond to what the classifier has learned. Some gradient-based methods, on the other hand, are equivalent when only considering **ReLU** activations in the hidden layers [171, 172], which is the case for EMOGI. **LRP** has reported good results in practice [151] and is faster than DeepLIFT [163], for instance. And although it was reported not to consider deeper layers of **DNNs** [209], the EMOGI model only uses relatively shallow architectures (see Table 5.1), alleviating that problem.

To recapitulate, **LRP** uses the weights of a trained neural network and propagates a relevance measure R back to the input space. Initially, the relevance $R^{(L)} = f(x)$ and is then redistributed through the network. The basic implementation supports fully connected and **CNN** architectures but for this work, **LRP** was adapted to **GCNs**. When applying **LRP** to **GCNs**, let us remember that each graph convolution layer does the following computation:

$$H^{(l+1)} = \sigma\big(LH^{(l)}W^{(l)}\big) \tag{5.7}$$

where $L = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ denotes the normalized graph Laplacian, $\widetilde{A}$ the adjacency matrix with added self connections, $\widetilde{D}_{ii}$ the degree matrix, W a learnable weight matrix and σ a non-linear function, such as the **ReLU** activation function (see Section 4.5.3 for details).

At the time when the project was implemented, no adaptations of gradient-based attribution methods to **GCNs** were known in the literature. Recently, however, two studies have adapted **LRP** to work with such models [210, 211] and a third article provides more theoretical insights [212]. The proposed method by Hu et al. [211] to interpret **GCNs** with **LRP** treats graph convolutional layers as two fully connected layers, applying the standard **LRP** rule (Equation 4.22) twice: once to decompose features (X) and weights (W) and once again to decompose the result and the graph Laplacian L. This results in the following equations for each layer:

$$\widetilde{R}_i^l = \sum_j \frac{\widetilde{x}_i^l w_{ij}^l}{\sum_i \widetilde{x}_i^l w_{ij}^l} R_j^{(l+1)} \tag{5.8}$$

where $\widetilde{X} = LH^{(l-1)}$. To then decompose the graph Laplacian and input data, the same rule is used again:

$$\widetilde{R}_i^l = \sum_j \frac{x_i^l L}{\sum_i x_i^l L} R_j^{(l+1)} \tag{5.9}$$

In this work, the deepexplain package [171] was used to compute **LRP** for genes of interest. Making use of the "input times gradient" rule, deepexplain directly computes the **LRP** values using the following equation:

$$R_i = x_i \cdot \frac{\delta^g S_c(x)}{\delta x_i} \tag{5.10}$$

where $S_c^g(x)$ corresponds to the output of the network at output unit (or neuron) c for an input data point x. g indicates a modified gradient rule for non-linearities ϕ (activation functions) denoted as: $g^{LRP} = \frac{\phi(h_i)}{h_i}$ where h_i corresponds to the activation of unit i prior to the activation function (see Figure 4.8 for a visualization of h_i). Equation 5.10 computes the relevance R_i for any unit i of the neural network. When we now only consider **ReLU** activation functions in a **GCN** model, we can immediately see that the non-linearities reduce to 1 for positive h_i and 0 for all other values for h_i. Therefore, the approach by Hu et al. is equivalent to the computations in this work.

In its original application for image classification, LRP computes a value for each pixel of an input image, indicating the importance of that pixel for the classification. Therefore, it returns for each input image a matrix, also called *relevance map*, of the same size of the original image, where each cell visualizes the relevance of a single pixel for classification [158, 166, 171].

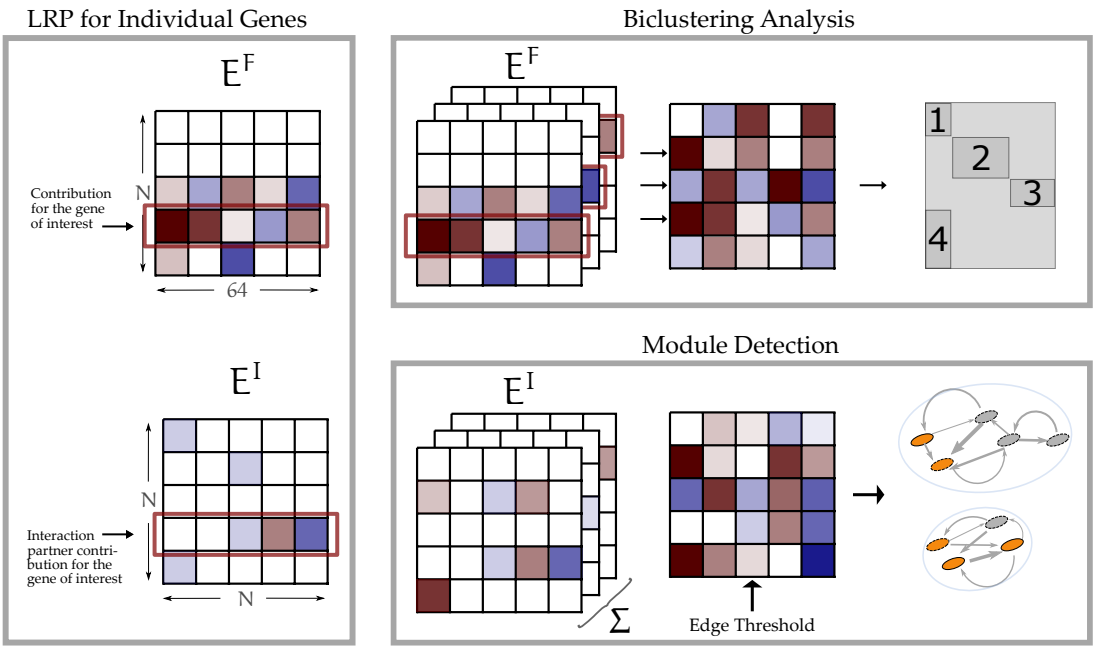


Figure 5.9: **Extraction of explanations using LRP.** Left panel: for each gene **LRP** returns 1) a feature importance matrix $\mathbf{E}^F$ containing the *omics* feature contributions from each cancer type to the gene of interest and 2) an interaction importance matrix $\mathbf{E}^I$ containing the PPI interaction contributions to the gene of interest. The right panel (top) summarizes how **LRP** and the obtained feature matrices for each gene $\mathbf{E}^F(g)$ are summarized to obtain the $\mathbf{E}^F_{total}$ matrix which is subjected to bi-clustering analysis (Chapter 7). The right panel (bottom) summarizes how **LRP** and the obtained interaction importance matrices for each gene $\mathbf{E}^I(g)$ are aggregated to obtain the $\mathbf{E}^I_{total}$ matrix which is then used for module detection (Chapter 7).

For the application presented here, the input to EMOGI consists of two matrices, namely the graph Laplacian L and the feature matrix X. Therefore, when **LRP** is used to find the most relevant *omics* features and **PPIs** for a gene of interest g, we obtain for that gene one matrix for the feature explanations, which is denoted $E^F(g)$ and one for the explanations of interaction partners, denoted $E^I(g)$. These matrices have the same shape as the input matrices for features and network, namely $(N \times 64)$ and $(N \times N)$, as each gene is characterized by 64 feature values and N corresponds to the number of genes in the **PPI** network (see Section 5.6 for details).
E^F and E^I can be obtained using:

$$E^F(g) = X \cdot \frac{\delta f(X, L)}{\delta X} \tag{5.11}$$

$$E^I(g) = L \cdot \frac{\delta f(X, L)}{\delta L}, \tag{5.12}$$

where both matrices X and L are treated as individual inputs to the **GCN**.
The intuition behind this is that **GCNs** aggregate features from neighboring nodes, resulting in a setting where the classification of a gene is not solely based on the features of that gene but also the features of surrounding genes in the **PPI** network. Therefore, explanations for genes are also not only based on the gene of interest but

also its surrounding genes in the network. Hence, both $E^F(g)$ and $E^I(g)$ are matrices instead of vectors as one would intuitively assume. In practice, the values of all rows (genes) in $E^F(g)$ and $E^I(g)$ are close to zero or very small except for the values corresponding to the gene of interest g and it can be easily seen from equation 5.12 that $E^I(g)$ is 0 for non-interacting gene products.

To answer the question about which cancer-type specific *omics* features mostly contribute to the classification of a certain gene, the row of $E^F(g)$ that corresponds to the gene of interest is extracted, resulting in a vector with 64 entries containing the importance of every *omic* feature in each cancer type (see Figure 5.9, left panel). The sum of the entries of this vector of feature explanations corresponds to the total multi-omics feature contribution for the gene of interest.
Similarly, to answer the question about the most important interaction partners for the classification of gene g, the row of $E^I(g)$ that corresponds to that gene is extracted (Figure 5.9, left panel). This yields a vector with N entries, each denoting the importance of each gene in the network to the classification of the gene of interest (g).
In Section 5.6, we saw that EMOGI is an ensemble method that uses 10 different **GCN** models to predict cancer-related genes. Consequently, all **LRP** computations are done for the 10 models independently and averaged to produce a final interpretation for a gene. This allows quantifying measures of uncertainty like the standard deviation or variance of explanations. $E^F(g)$ and $E^I(g)$ are assumed to be averaged matrices across the 10 models for the remainder of the section.

For the bi-clustering analysis that will be presented in Section 7.2 the vectors of feature explanations for all genes are simply stacked on top of each other to create a matrix of shape $(N \times 64)$, denoted as E^F_{total}, that can directly be subjected to the biclustering algorithm (Figure 5.9, right panel).
The module analysis — presented in Section 7.3 — aims at finding core sub-networks which represent the most important parts of the interactome used by EMOGI for cancer gene classification. In order to understand how **LRP** values for gene interaction partners are used to identify such important modules, we need to keep in mind that EMOGI mostly uses a two-layered **GCN** (Section 6.2.1 will show that only few graph convolutional layers are required for good performance) which learns and propagates features from a first-order gene-gene interaction neighborhood to a second-order neighborhood. This means that higher order interactions, and not only direct interactions between genes, contribute to the classification
To capture core interactions that are repeatedly important for the classification of multiple genes, and therefore identify cancer-related modules, it is not sufficient to extract, for a given gene g, the row of $E^I(g)$, corresponding to the importance of its direct interactors. Instead, also the indirect contributions from the **PPI** graph have to be taken into consideration.

To achieve that, the N interaction explanation matrices $E^I(g)$ were collected for all genes. Those matrices were subjected to an element-wise summation to obtain the matrix E^I_{total}, where the value of each cell represents the total importance of a certain **PPI** for the classification of the input genes (Figure 5.9, right panel). The higher the value in a cell of E^I_{total}, the more important was the interaction globally, often for many genes.

Next, min-max normalization is applied to each row of E^{I}_{total}, such that every gene contributes the same total amount of **LRP** to the matrix, thereby removing biases from genes with high node degree in the **PPI** network. The final normalized matrix E^{I}_{total} is then converted to a directed, weighted graph, where nodes correspond to the genes from the original **PPI** network and edge weights correspond to the entries of the normalized E^{I}_{total}, i.e. to the importance of each interaction in the EMOGI model. Finally, a threshold was selected and all edges with weights below that threshold are removed to obtain a sparse graph, which is then subjected to the **Strongly-Connected Component (SCC)** calling algorithm (see Figure 5.9 and the results in Section 7.3).

5.9 IMPLEMENTATION & CODE AVAILABILITY

EMOGI was implemented in Python using Tensorflow [177]. The basic implementation of **GCNs** [133] was extended and for the **LRP** computation, the deepexplain package [171] was used.
All source codes to train the EMOGI model or to reproduce the results are freely available on `https://github.com/schulter/EMOGI`. This repository also contains manifest files that can be used to download **TCGA** data using the *GDC Data Transfer Tool*. The trained multi-omics models for all six **PPI** networks can be downloaded from `https://owww.molgen.mpg.de/~sasse/EMOGI/`.

5.10 SUMMARY

In this chapter, we saw how the EMOGI model is constructed. The **GCN** method from [133] was extended and adapted to a pan-cancer gene stratification setting. The multi-omics data sets from **TCGA** were individually preprocessed for 16 different cancer types to derive pan-cancer multi-omics feature vectors for ten thousands of genes. Furthermore, 6 different **PPI** networks were collected and preprocessed to fit with the **GCN** model. Class imbalance was taken into account through incorporation of a weighted loss function and the EMOGI model was heavily regularized using dropout and norm penalties. To increase the robustness and to quantify uncertainty in both predictions and explanations for genes, multiple **GCN** models were trained as an ensemble. The high number of **HPs** were optimized using a grid search in a **CV** scheme. And finally, EMOGI predictions can be explained using the **LRP** framework, yielding contribution scores of multi-omics features across cancer types as well as contributions of individual interaction partners in the **PPI** network.

6 VALIDATING THE EMOGI MODEL

Chapter 5 gave an overview of how the EMOGI model was constructed. This chapter will evaluate the model predictions using a variety of strategies. First, we will see how EMOGI captures patterns in simulated data, both from a network and feature vectors. This demonstrates the model's capability to profit from non-redundant and heterogeneous data types.
Because it is not entirely clear if the catalogue of cancer driver genes is complete [19, 25, 31, 198] and because more and more diverse mechanisms of how cancer malignancies hijack the normal cell regulations are being discovered (as discussed in Section 2.2), there is no gold standard of cancer genes available. Therefore, EMOGI was validated on a broad variety of sets of putative cancer driver genes and **LRP** for well-studied genes was used as an additional validation. The reasoning behind the latter being that if the features important for the classification of a well-studied gene match with the literature and experimental data from mouse models or other *in vivo* data, EMOGI can be trusted to successfully capture biological signals.
EMOGI does not only integrate data from different *omics* levels but also integrates different cancer types in a pan-cancer analysis to amplify signals present in multiple cancers. It was shown previously that rare mutations can be detected reliably with pan-cancer studies when cancer specific approaches fail [30] and a similar rationale holds true for DNA methylation and gene expression data [37, 61, 67], as introduced in Section 2.2.2.

The robustness to noisy data as well as the benefit of integrating multiple *omics* levels was validated using perturbation experiments where either the **PPI** network or the feature vectors were randomly perturbed prior to model training. Furthermore, by leaving out entire data sources, such as **SNVs**, **CNAs**, DNA methylation or gene expression data, the impact of data types was assessed. The drop in performance of the reduced model can be used to assess the importance of the perturbed or removed feature and gives first novel insights into cancer biology.

6.1 SIMULATED DATA

Validating a model that uses feature vectors and a graph structure as sources of information to classify the nodes in the graph is not trivial. This is because in real-world data sets, artifacts and biases instead of real biological features might govern the classification. Simulated data can help to ensure that the learning is working in the expected way. Section 2.1.3 has shown that proteins organize in complexes and pathways. Biological networks in general and **PPI** networks in particular were shown in the past to form densely connected modules that connect to other regions through only relatively few nodes [213].
The goal of this first simulation experiment was to see whether EMOGI can truly capture topological properties of biological networks and feature vectors simultaneously.

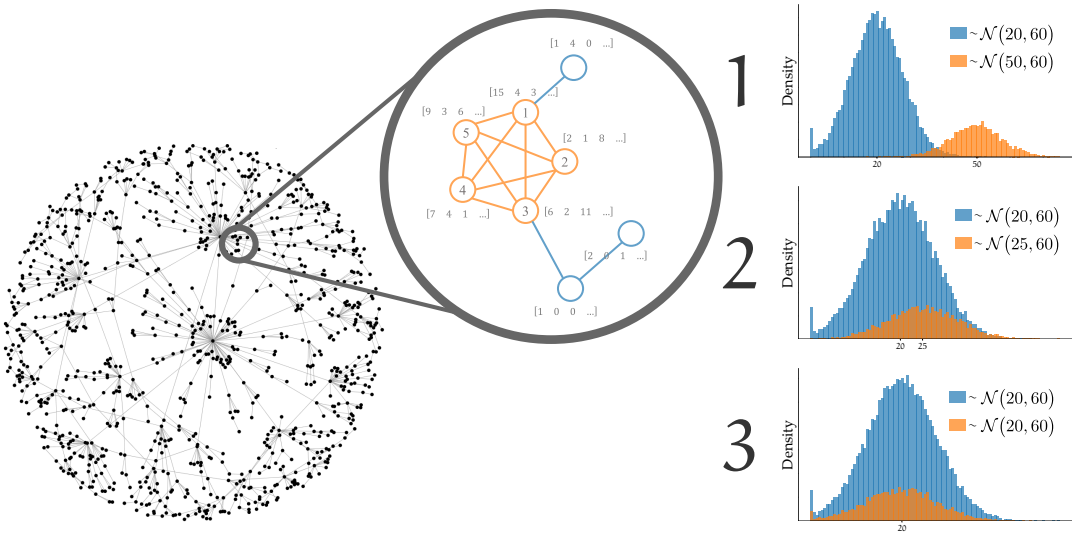


Figure 6.1: **Simulated network and features.** A random network with 1053 nodes and 2200 edges was generated and 38 cliques of size 5 were inserted at random positions. Feature vectors $x_i \in \mathbb{R}^{24}$ were drawn from normal distributions. Feature vectors for the 190 ($38 \cdot 5$) clique nodes were drawn from the orange distribution while the other nodes received feature vectors drawn from the blue distribution. Randomized network image taken from `https://flic.kr/p/dQyBpn`.

To that end, a binary supervised classification problem was formulated similarly to Figure 4.3 in Section 4.1.

A graph G that resembles a biological network was simulated and graph motifs were implanted into G. Figure 6.1 shows how a clique (a subgraph where every node is connected to every other node) was implanted into a random network by adding edges to the graph. For the generation of such networks, the tool NetSim[1] was used. Thirty-eight cliques of size 5 were implanted at random positions into a network with 1,053 nodes while each clique was at least two nodes away from all other cliques.

Feature vectors were simulated through Gaussian random numbers drawn from two different distributions. The first distribution (marked in blue in Figure 6.1) was used to generate feature vectors for the nodes outside of the graph motifs while the orange distribution was used for the nodes within the graph motifs. All nodes were split into train and test sets and EMOGI was trained on the simulated data. For that, a non-optimized set of **HPs** was used and the performance of EMOGI was compared to that of a logistic regression.

The three different setups represent situations where the node features contain varying degrees of information regarding the classification task to correctly distinguish clique nodes from the rest of the network. In the first simulation, the node features are very informative and should be more important than the network topology. In the second simulation, node features are only slightly different for the two classes and the network topology could be helpful to distinguish the two classes. And for the third simulation, the node features are drawn from the same distribution and only the network topology is informative to classify the nodes correctly.

1 `https://github.com/schulter/NetSim`

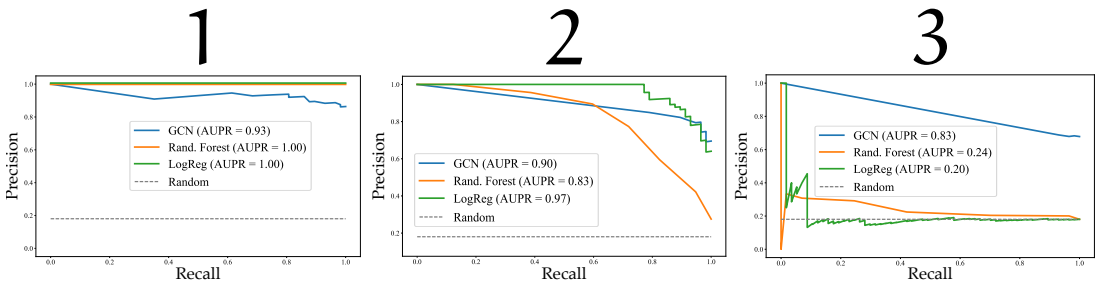


Figure 6.2: **Simulation results on a random network with cliques.** For all three scenarios depicted in Figure 6.1, a precision-recall curve depicts the performance of EMOGI and two other supervised classification methods. The random forest is a non-linear method while logistic regression is a linear algorithm.

Figure 6.2 depicts the performance on the test set for the three different setups. In the first, not only EMOGI but also a logistic regression (introduced in Section 4.1.1) can easily distinguish between the two classes. The logistic regression achieves an area under the precision-recall curve of 1.0 which means that it classifies every data point in the test set correctly.
The second scenario puts the two distributions already much closer to one another and the overlap between the two distributions is significant. Due to the 24 values per feature vector, however, the nodes can still be fairly well classified by a logistic regression. EMOGI still performs slightly worse compared to the logistic regression in this setting but some points are missed. This is probably because the logistic regression only learns to assign higher features to the positive class and lower features to the negative class.
In the third setting, both distributions are equal and only EMOGI is able to classify the nodes correctly while the logistic regression fails and predicts nodes at random. The performance of EMOGI, however, is lower compared to the first two situations. This shows that EMOGI is capable of using both sources of information simultaneously. Furthermore, protein complexes often correspond to cliques in **PPI** networks [214–216] and hence their successful prediction without explicitly searching for them is a useful property. This behavior also demonstrates that EMOGI can learn patterns solely based on the graph topology.

6.2 PERFORMANCE EVALUATION

Knowing that EMOGI can benefit from network and feature information, especially in cases where the information is non-redundant, encourages applying the method to the pan-cancer and multi-omics data set discussed in Chapter 5. However, to apply such a complex non-linear classification method, an optimized set of **HPs** is needed and the algorithm has to be exhaustively validated and tested on multiple sets of putative cancer genes.

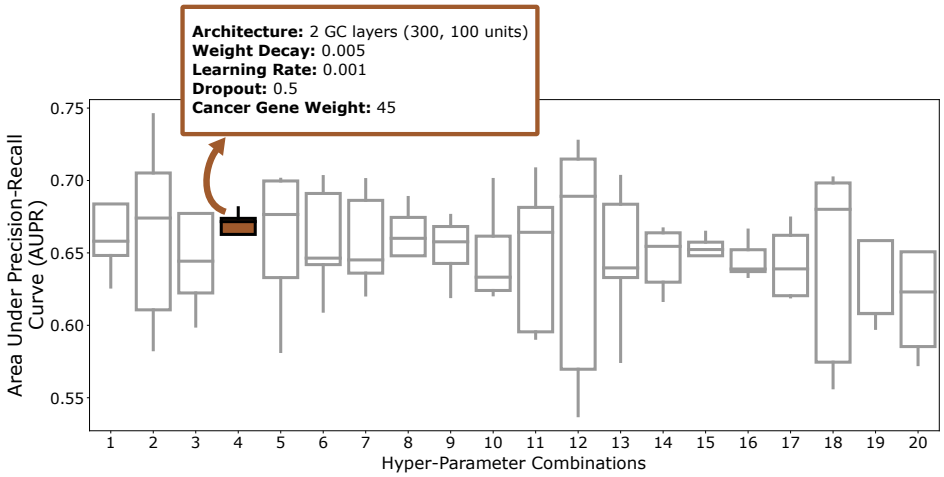


Figure 6.3: **Hyper-Parameter optimization of the EMOGI model.** The combination of **HPs** that yielded the 4^{th} highest average **AUPRC** was selected because it resulted in the best compromise between robustness and performance.

6.2.1 *Finding Hyper-Parameters for EMOGI*

As introduced in Section 5.7, the EMOGI algorithm contains several hyper-parameters that have to be set and heavily influence the performance of the model. On top of the classical parameters of literally all deep learning models, a weight for the positive class (a scalar factor by which the loss of known cancer genes is scaled to increase their importance) is required to account for class imbalance. To select the best **HPs** for EMOGI, a grid search was conducted using 5-fold **CV**. As introduced in Section 4.2.3, **CV** splits the data into n equally sized subsets. This is followed by n iterations in which $n-1$ splits are used for training and the ith split is used for validation to compute the classifier's performance.
A grid search over a reasonable number of values for each **HP** was conducted as explained in Section 5.7. EMOGI was optimized on the **CPDB PPI** network for every possible combination of those values, corresponding to 288 combinations of **HPs** in total. Figure 6.3 depicts the 20 highest-performing **HP** combinations, evaluated using the **AUPRC**. The best combination of **HPs** was selected as the combination that yielded the best median **AUPRC** values after 2000 training epochs but that is also robust, i.e. has low variance across **CV** folds (the 4^{th}-best combination in Figure 6.3). This resulted in a dropout rate of 0.5, a learning rate of 0.001, multiplication of the loss for positives by 45, a weight decay of 0.005 and two graph convolutional layers with 300 and 100 filters, respectively. This is in line with previous studies that reorted over-smoothing with higher numbers of layers [136, 142].

6.2.2 *Training EMOGI on Multi-Omics Data*

EMOGI was trained on a high confidence set of cancer and non-cancer genes based on multi-omics features and various **PPI** networks from publicly available databases

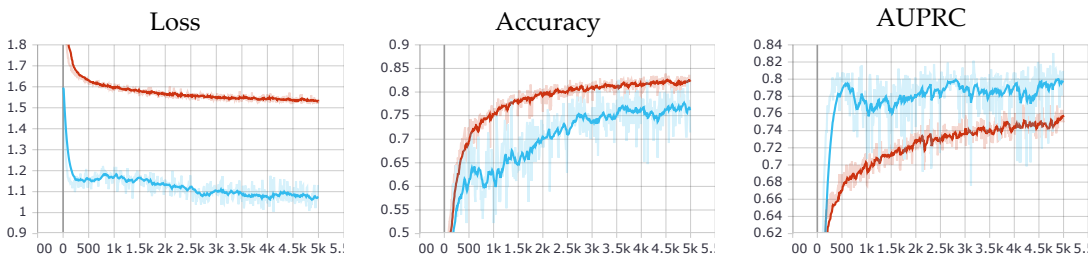


Figure 6.4: **Performance evolution over training time (epochs).** Depicted is the evolution of the loss, accuracy and **AUPRC** (see Section 4.2.2 for a description of the metrics) over the time of the training (7,000 epochs in total), averaged over 10 different **CV** runs on the **CPDB PPI** network. The blue graph denotes the test set performance and the red graph denotes the training set performance for the different metrics. Accuracy is measured using a threshold of 0.5.

(as explained in detail in Section 5.2), in order to assess whether the method's performance is consistent across different **PPI** networks (see Section 5.1 for an overview of the training process).

During the optimization procedure, various performance metrics were measured to monitor training progress and detect potential overfitting. Figure 6.4 depicts the computed loss, accuracy and **AUPRC** values as a function of epochs (the training time), averaged over the 10 different **CV** runs for the **CPDB PPI** network. No overfitting can be detected in this setting and the training loss converges. The high difference between training and test loss comes from dropout which is only used during training but not during inference.

After successful training using the **CPDB** graph, EMOGI predicts 4,522 genes to be associated with cancer using a threshold of 0.809 for the predictions. The threshold is selected based on the intersection between precision and recall, as depicted in Figure 6.5 on the left side. Both, precision and recall are regarded as functions of a specific threshold (see Section 4.2 for details). The intersection between the two metrics corresponds to a threshold (on the x-axis) that optimally balances precision and recall which is desired in this application. To select genes for further hypothesis-driven studies, a higher threshold is probably recommended that produces more precise cancer gene predictions.

The EMOGI model further recovers 89% of **KCGs** and 47% of **CCGs** to be associated with cancer malignancies (Figure 6.5, right side).

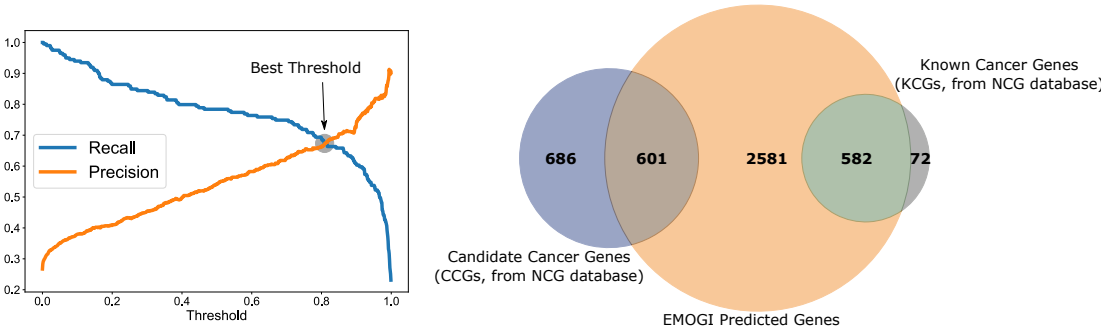


Figure 6.5: **Overlap of EMOGI's positive predictions with KCGs and Candidate Cancer Genes (CCGs) on the CPDB PPI network and cutoff selection.** To quantify an overlap between gene sets, a threshold on the EMOGI score had to be chosen.The cutoff was selected such that it balances precision and recall optimally as depicted on the left. Recall decreases when the cutoff increases because fewer genes are considered positive predictions. Precision, on the contrary, increases when the cutoff increases because the predictions are more and more conservative and only high-confidence predictions are considered for a high cutoff value. The intersection between the two metrics therefore represents a good value for the cutoff. The optimal selected threshold is 0.809 for EMOGI on the **CPDB PPI** network.

6.2.3 *Performance on Validation Sets*

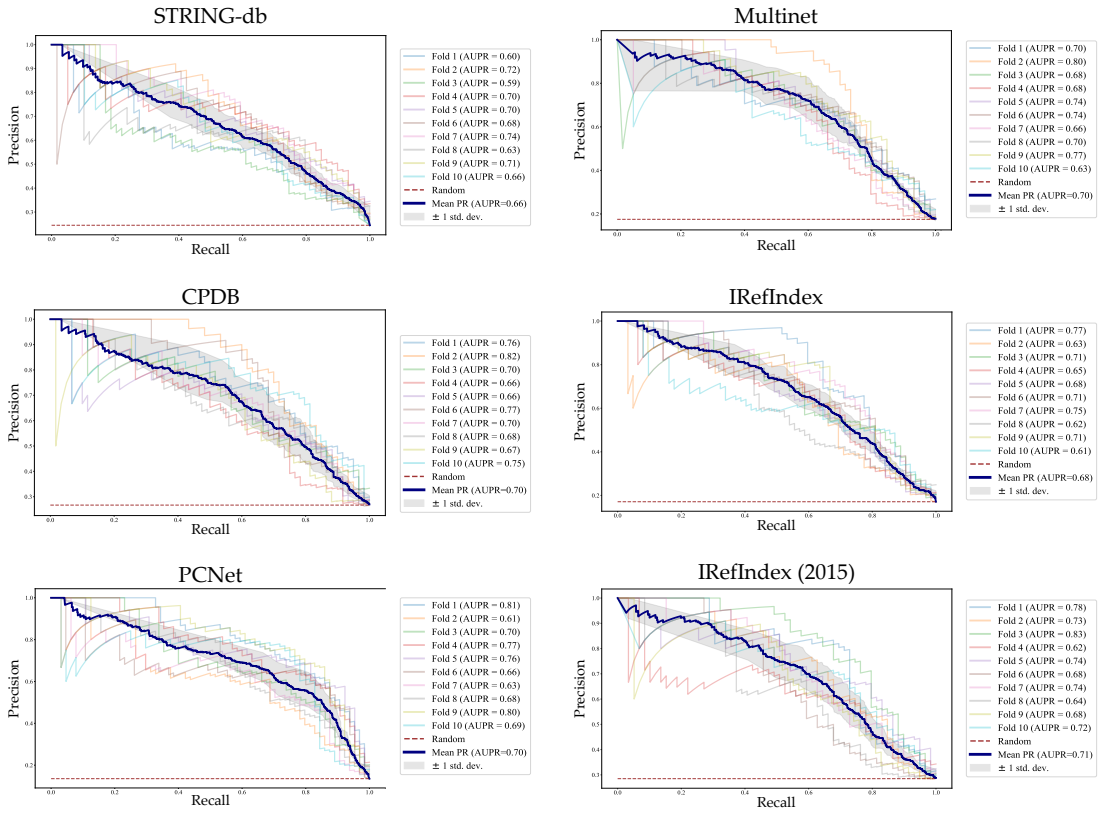


Figure 6.6: **Test set performance of EMOGI.** EMOGI was trained on the pan-cancer data using 10-fold **CV**. This training process produces 10 different models whose performance is depicted here. EMOGI was also trained on 6 different **PPI** networks to assess robustness. Each of the 10 folds of the **CV** is depicted in the precision-recall curves. The thick blue line denotes the average performance and the gray shading denotes ± 1 standard deviation

Next, the performance of the successfully trained EMOGI models for all six **PPI** networks was analyzed on the 10 different validation sets. As depicted in Figure 6.6, EMOGI exhibited a range of **AUPRC** values between 66% and 71% for six different PPI networks computed on the validation sets.
The averaged performance (micro-averaged PR curves were used) is very similar across **PPI** networks with STRING-db performing worst in this context. Interestingly, the older IRefIndex network from 2015 yields the best performance of EMOGI despite the network having the least interactions. The reasons for that behavior are not clear but a study bias [217] could explain the phenomenon. Study bias refers to an artifact in the data that originates from an unequal interest in proteins and genes. Canonical cancer genes, for instance, are investigated much more often in **Y2H** or other **PPI** screens, thereby producing a network where previously identified disease genes exhibit higher node degree and other characteristics [124–126, 218]. And indeed, node degree is 5 times higher for the gene sets used to train EMOGI than for other genes. The **CPDB PPI** network, on the other hand, was previously reported to be useful to identify disease genes in a variety of diseases [190] but EMOGI seem to be unable to profit from that. To further shed light into this issue but also to assess the performance of EMOGI on the test in comparison to established cancer gene prediction methods, several other methods and tools were used on the EMOGI training and test data sets.

6.2.4 *Existing Methods for Cancer Gene Prediction*

EMOGI was benchmarked against a large variety of **ML** methods for the prediction of cancer genes in order to validate its performance on a test set that was held out during training and **HP** optimization. The compared methods represent the state-of-the-art from both, the **ML** community and the cancer genomics field. They can be distinguished based on whether they use the network topology, multi-omics features of genes or both. Furthermore, some methods were specifically tailored for the prediction of cancer genes (briefly introduced in Section 2.2.2) and a thorough comparison with them is essential to prove that integration of heterogeneous data types is an advantage.

Random Forest Model

A Random Forest is a non-linear classifier that uses ensembles of decision trees for classification. It fits multiple such trees in a greedy optimization scheme and makes use of a majority vote between the trees to decide on a class.
A Random Forest classifier to predict cancer gene was trained on the multi-omics features only, namely genomic (from **SNVs** and **CNAs**), epigenomic and transcriptomic features, regardless of the **PPI** network using the *RandomForestClassifier* function, with default parameters, from the *scikit-learn* python package [219]. The algorithm is a generic, supervised **ML** method that used the same set of **KCGs** and non-cancer genes as EMOGI. The result of the Random Forest consists of a probability for every gene to be associated with cancer which was consequently used to derive **AUPRC** values.

DeepWalk

This method computes a node embedding in an unsupervised fashion from the network alone and does not incorporate node features or labels. DeepWalk constructs a vector representation of each node, trying to preserve the topological features of that node such that nodes with similar topologies are close to one another in the learned representation. It works by conducting small random walks on the graph iteratively, starting at each node in the network at a time. These walks explore the local structure around the node for which the representation is to be computed and are interpreted by DeepWalk as sentences of natural language. The algorithm then iterates over the random walks starting at node i and optimizes the node embedding to maximize the probability that the other nodes of the random walk are close to node i in the learned representation. This method, called skip-gram model, was very successfully used in natural language processing [220]. Here, DeepWalk is used to compute a vector representation of the nodes and a **Support Vector Machine (SVM)** with radial basis function kernel is applied to that representation to classify between cancer and non-cancer genes. The use of an **SVM** as was suggested by the authors of DeepWalk in a personal correspondence. Training and test sets for training of the **SVM** were identical to those used for training EMOGI. To find optimal **HPs** for DeepWalk, a grid search was used over reasonable ranges. Optimal values for the length and number of the random walks, the size of the vector representation and the window size were determined that way.

GCN Without Node Features

The same implementation of **GCNs** was used but the feature matrix was replaced by a vector of 1s, as suggested by the authors [2]. To obtain reasonable hyper-parameters, a small grid search was conducted and it was found that the **HPs** used by the complete EMOGI model worked well also in this case. Therefore, EMOGI was trained based on the **PPI** network topology using an architecture of two graph convolutional layers with 300 units in the first layer and 100 units in the second layer, a dropout rate of 0.5, weight decay of 0.005, a learning rate of 0.001 and a cancer gene weight of 45.

PageRank

The PageRank algorithm, originally developed by Google to rank web pages in their search engine result, conducts **Random Walk with Restarts (RWRs)** on a network to identify highly influential nodes [221]. The PageRank algorithm outputs a probability distribution used to represent the likelihood a certain node in a network will be reached by starting randomly somewhere in the network. It was used here to rank genes according to their likelihood of being cancer genes based on the **PPI** network, irregardless of their *omics* features. An implementation of PageRank from the *NetworkX* python package with a restart probability of 0.3 was used and node probabilities were used to compute **AUPRC** values.

2 https://github.com/tkipf/gcn/issues/10

DeepWalk + Features

EMOGI was also benchmarked against a combination of DeepWalk embeddings alongside the multi-omics feature matrix computed for the 16 cancer types. For that, the DeepWalk embeddings of the **PPI** network were concatenated with the *omics* features for each gene. Next, a Random Forest (RF) classifier (see section on Random Forests above) was trained on the combined matrix using the same labeled data and the same training and test splits as EMOGI. A Random Forest was chosen because the *omics* values and DeepWalk embeddings are not likely to span the same feature space but RFs are known to work well with differently scaled data. Again, the Random Forest implementation from the *scikit-learn* python package was used using default parameters.

HotNet2 Diffusion

Network propagation amplifies a biological signal based on the assumption that neighboring genes in an interaction network are more likely to share the same phenotype [69]. The algorithm of Network Propagation forms the basis of HotNet(2) [41, 68] and hierarchical HotNet [42], where an average mutation score across cancer types is associated with every node in the **PPI** network and propagated to nearby nodes in an iterative manner using **RWR**, similarly to the PageRank algorithm. Here, the HotNet2 implementation of the **RWR** procedure and their precomputed heat scores for the genes were used. The diffusion process converges in a steady-state probability distribution from which **AUPRC** values can be computed, similar to the PageRank application.

MutSigCV

MutSigCV is one of the most popular tools to identify cancer driver genes [40]. It prioritizes highly mutated genes, i.e. genes with a mutation frequency which is higher than what expected by chance, based on a background model which takes into account both gene sequence composition and gene length. Here, MutSigCV was used to predict cancer genes from one type of *omics* data only, the mutation rate. Gene-associated q-values for each of the **TCGA** studies were computed separately and averaged over all 16 cancer types to produce gene scores. The $-\log_1 0$ was used to compute **AUPRC** values, similar to [41].

20/20+

This method is a widely used machine learning method that predicts cancer genes based on mutations. Tailored specifically to the problem of cancer gene prediction, 20/20+ uses a random forest alongside a ratiometric approach to predict oncogenes and tumor suppressor genes [31].
Here, the tool was applied to the MAF files for all 16 cancer types used in this thesis which were obtained from **TCGA**. Specifically, all 16 MAF files were concatenated using custom code. To be used with 20/20+, a liftover of the mutations from hg38 back to the hg19 reference genome had to be computed because 20/20+ is not compatible with the newer genome version. Next, 20/20+ was applied to the hg19 mutation

coordinates from the concatenated MAF file, as indicated in the documentation for pan-cancer applications [3]. Finally, the "driver score" was extracted for all genes to compute **AUPRC** values.

6.2.5 *Comparison to Existing Cancer Gene Prediction Methods*

The comparison to the aforementioned methods was done using the test set that was not shown to any of the methods during training and the **AUPRC** value on the test set was used as a metric. **AUPRC** is the most appropriate metric for this comparison because it is independent of the choice of a specific cutoff to determine if a gene is associated with cancer or not and, in addition, accounts for class imbalance.
If competing methods required training, the training set of EMOGI was used for that. PR curves do not require methods to output probabilities but can base the curve on any continuous score. EMOGI was compared to baseline methods such as a Random Forest classifier, the PageRank algorithm [221] and the DeepWalk method [222], as well as other tools specifically developed for cancer gene prediction, such as MutSigCV and network propagation-based methods such as HotNet2 [41, 69]. Furthermore, the capability to predict cancer genes of several established node metrics, such as degree, core, betweenness and the clustering coefficient were investigated and compared to EMOGI for the **CPDB PPI** network, as depicted in Figure A.5.

3 https://2020plus.readthedocs.io/en/latest/tutorial.html#pan-cancer-analysis

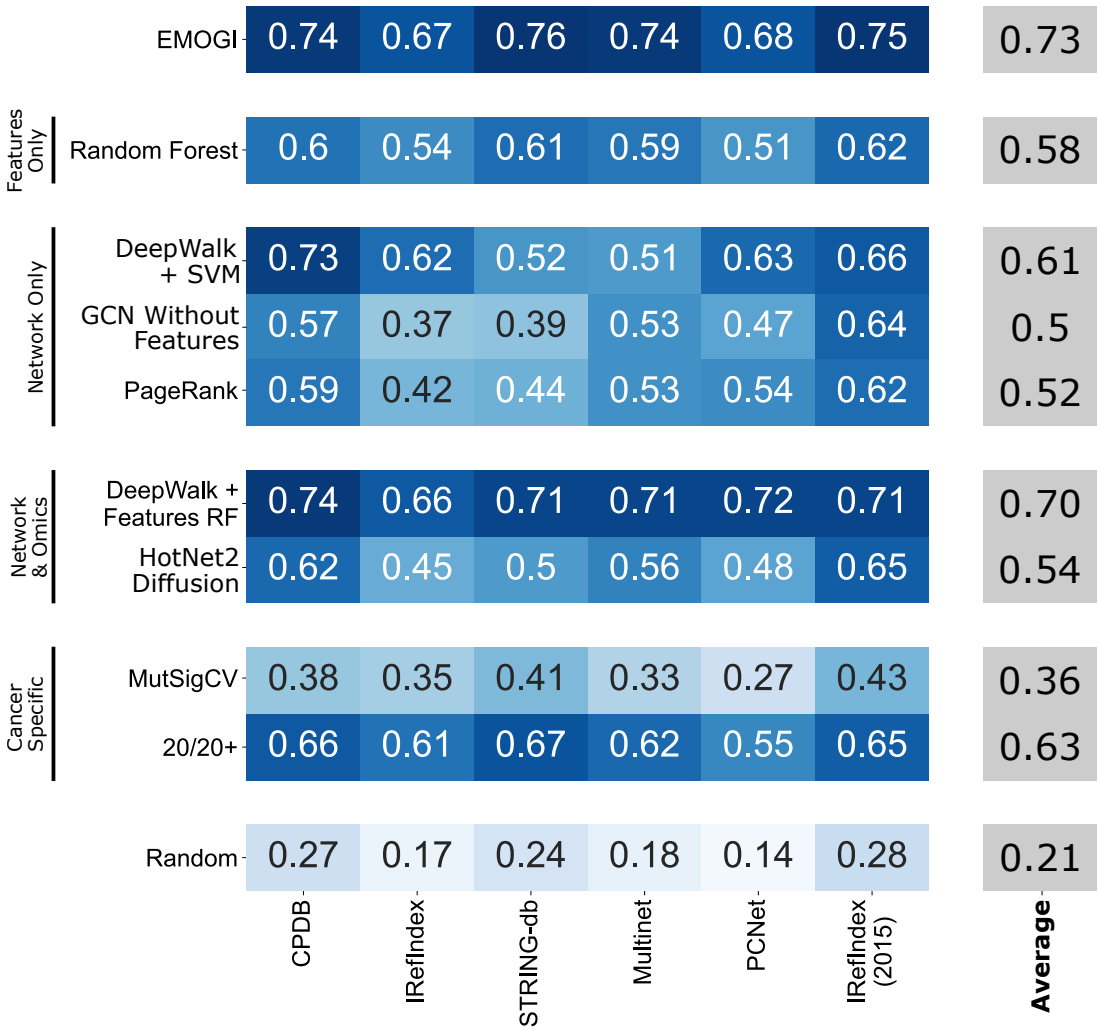


Figure 6.7: **Performance comparison between EMOGI and other methods.** EMOGI's performance on the test set was evaluated and compared to other methods across 6 different **PPI** networks. **Area under the Precision-Recall Curve** for different prediction methods was computed on a test set of known cancer and non-cancer genes which was held out during model training and **HP** optimization. Dark blue cells in the heatmap correspond to high performance, i.e. high **AUPRC** values, while light blue ones correspond to lower performance.

Random forests operate on feature matrices and hence, in this setting it makes use only of the gene features while discarding the **PPI** network topology. PageRank, on the other extreme, is a popular algorithm which uses only the information encoded in the **PPI** network for classification, while discarding node features. It works by regarding random walks on a graph as a Markov chain that reaches an equilibrium after a certain number of steps. PageRank computes this equilibrium (often called steady-state distribution) in a closed form and thus finds influential network nodes. The DeepWalk method also operates only on the **PPI** network and was chosen for comparison because it captures the topology of networks beyond metrics like node degree or betweenness [222]. Both, DeepWalk and the PageRank algorithm are un-

supervised **ML** methods. To assess the impact of supervision (or semi-supervision) in the context of cancer gene prediction, EMOGI was compared further to a **GCN** without features. This also uses the implementation from [133] but replaces the multi-omics features with an uninformative vector of 1. EMOGI was also compared to the HotNet2 diffusion process and a custom approach that couples DeepWalk embeddings with multi-omics features. The former benefits from both feature and network information through the use of **RWRs** and has been successfully used in the last few years to identify cancer genes, as well as network modules of cancer genes [41, 69, 190]. The latter uses the learned DeepWalk embeddings and concatenates them with the multi-omics feature matrix X. The resulting matrix is then used by a Random Forest to classify cancer-related and cancer unrelated genes using the EMOGI training data. The two methods profit from both, multi-omics features and **PPI** network topology and provide powerful baselines.
Lastly, EMOGI was benchmarked against two popular tools that are tailored to predict cancer genes, namely MutSigCV [29] and 20/20+ [31]. Both operate only on mutation frequency features but while MutSigCV find genes that are significantly more often targeted by **SNVs**, 20/20+ makes use of a ratiometric approach.
Figure 6.7 depicts precision-recall curves for EMOGI and the aforementioned other methods. The performance of EMOGI is in part higher compared to Figure 6.7 because the mean prediction across all 10 models on the test set was used instead of the evaluation on the different **CV** validation sets.
For all six **PPI** networks, EMOGI outperformed all other methods on average by at least 3% **AUPRC**. The custom implementation using DeepWalk embeddings as well as multi-omics feature vectors performs comparable on the **CPDB PPI** network and even outperforms EMOGI on PCNet but then does worse than EMOGI on all other networks. All other methods achieve substantially lower **AUPRC** values compared to EMOGI. Interestingly, both DeepWalk variants perform best on the **CPDB PPI** network, a resource which is known to enhance the computational prediction of true disease genes compared to other **PPI** networks [190]. In line with that observation, DeepWalk alone — operating only on the **PPI** graph structure — performs by far best on the **CPDB** network. Another interesting observation is the generally high performance of methods that only make use of the **PPI** network. It indicates that modern **PPI** networks have encoded the main properties of known cancer genes in their topology. This can either be due to a study bias where known cancer drivers such as *KRAS* or *TP53* are much more extensively studied than other genes and hence have more interaction partners in the network [217]. Or it can be that our filtering procedure for the non-cancer genes discards central genes in the **PPI** network, thereby biasing the test set, or a combination of both. Evidence for that is a 5-fold higher node degree for the positive set of known cancer genes compared to the negatives (non-cancer genes), a correlation between node degree and essentiality in loss-of-function screens (introduced in Section 7.1.3) and a higher node degree for all cancer gene sets, including novel predictions from EMOGI (introduced in Section 7.1), depicted in Figure A.4. However, node degree alone — or any other network metric such as betweenness centrality or clustering coefficient, for that matter — does not explain the high performance of DeepWalk (depicted in Figure A.5). All of these metrics perform considerably worse on the test set and similar observations with another computa-

tional method were made in [44].
Interestingly, DeepWalk is unable to distinguish cancer from non-cancer genes on the Multinet network. This can be due to the fact that Multinet also contains additional regulatory interactions and therefore does not only reflect **PPIs**. Because DeepWalk projects nodes close to one another that occur in the same random walk, cancer pathways are expected to be close in the projection and hence, the SVM is expected to identify those pathways reliably.
This leads to a fundamental problem in many scientific applications of supervised machine learning. The labeled data is imperfect and therefore, test set performance must not be the only criterion to rely upon.

6.3 PERFORMANCE ON INDEPENDENT GENE SETS

To better understand how consistent and robust EMOGI and the other methods perform in correctly recovering cancer genes, and whether performance is biased towards a specific dataset and/or **PPI** network, the performance of all investigated methods was assessed on four other sets of annotated cancer genes which were treated as additional independent test sets.
The first set is represented by candidate cancer genes from the **NCG** which are non-overlapping with the **KCG** set used for training of EMOGI [194]. The second set comprises a list of genes from the OncoKB database [193], a manually curated dataset of cancer mutations annotated according to validated oncogenic effects. Only high-confidence cancer genes, i.e. with evidence from more than three sources, were included in this set. The third set comprises a list of literature-curated cancer genes from the ONGene database [192] and the fourth set a list of high-confidence cancer driver genes compiled using different computational tools [198].

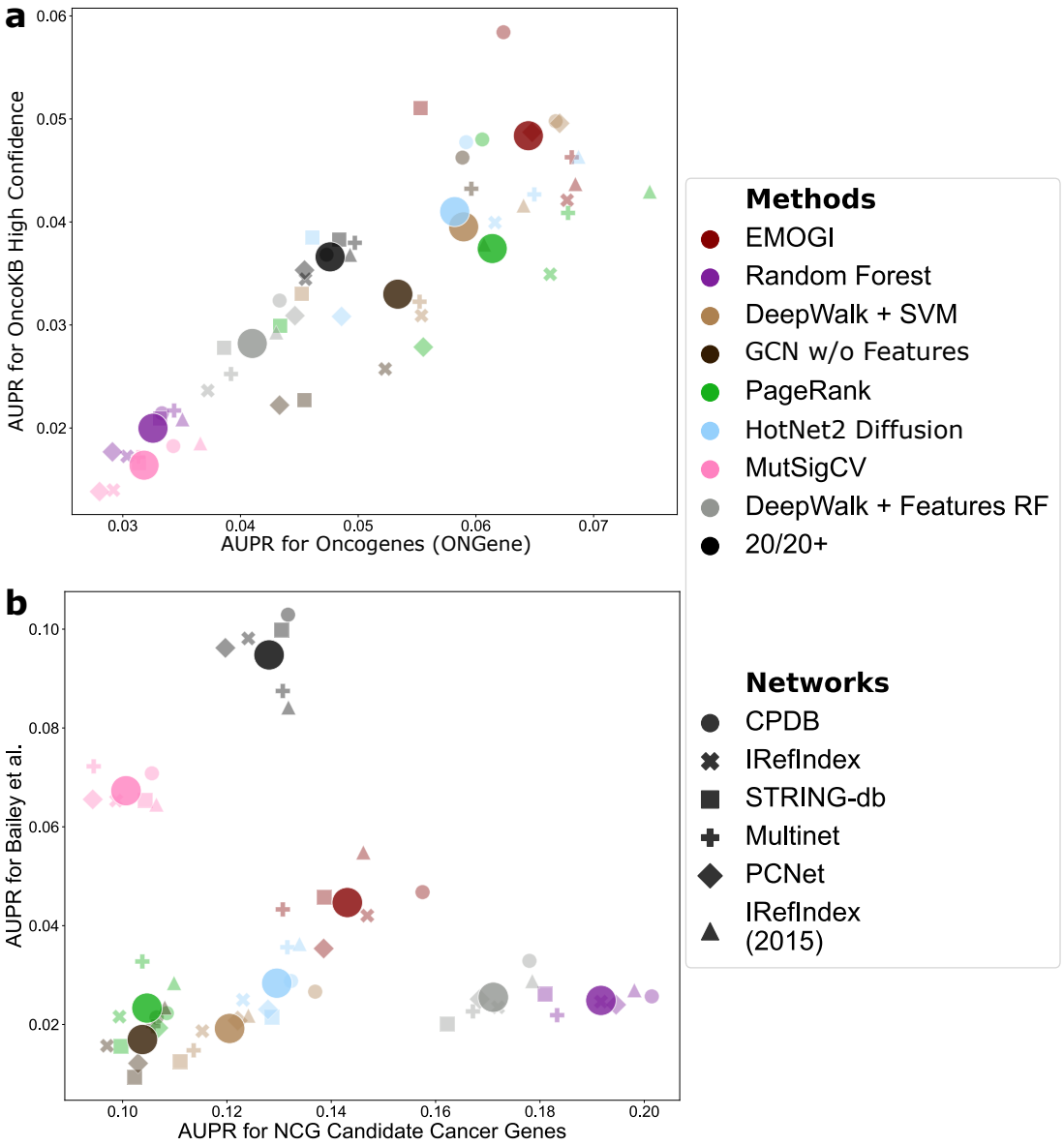


Figure 6.8: **AUPRC-AUPRC scatterplot comparing performances between methods on independent data sets.** The performance of each tool is measured using **AUPRC** values. Only hits in the gene set are considered true positives. All genes that were present in EMOGI's training or test sets were removed to ensure unbiased comparison. Each method and **PPI** network correspond to a point in the plots. The color indicates the method and the shape of the point indicates the **PPI** network. Points in the upper right corner exhibit higher **AUPRC** values for both gene sets. **a** Oncogenes from the ONGene database [192] and cancer genes from OncoKB [193] were used for a comparison. **b** A computationally derived cancer gene set from Bailey et al. [198] as well as the **Candidate Cancer Genes** from the **NCG** was used for a comparison of methods for cancer gene prediction.

Figure 6.8 shows the performance of all investigated methods. To compute AUPRC in this setting, hits in the gene set were counted as true positives and all other predicted cancer genes not contained in the set were considered false positives. This

resulted in much lower **AUPRC** values for all methods.
On the two sets of curated cancer genes from OncoKB and ONGene, EMOGI performed consistently better than other methods, although the performance of all tools (except the ones based on gene features only) differed vastly between **PPI** networks. On the candidate cancer genes from the **NCG**, a Random Forest classifier was the best performing method. The Random Forest, however, did not achieve high **AUPRC** on any of the other cancer gene sets. Because the candidate cancer genes are derived to a large proportion from **TCGA** data [194], they show highly differential gene expression and DNA methylation and are also often mutated, explaining why the performance of the random forest is so high. On the computationally derived data set from Bailey et al., the methods designed for cancer gene prediction (MutSigCV and 20/20+) performed best. This is reasonable because these genes represent the most highly mutated genes from cancer screens and have been proposed based on predictions by MutSigCV and 20/20+ [198].
While the performance of feature-only based methods, such as Random Forest, MutSigCV and 20/20+, was not stable and highly depended on the analyzed data set, EMOGI outperformed all the other network-based methods consistently on all four independent data sets, indicating that the method is robust across different types of data.
The evaluation of the different methods on six different **PPI** networks shows that **CPDB** represents a highly informative network with respect to the task of identifying cancer driver genes. Evidence for that is the high performance of network-only methods on the test set for **CPDB** (see Figure 6.7) and the high performance of that **PPI** network on all four independent data sets for various methods (see Figure 6.8) Therefore, the rest of the analyses will focus on **CPDB** as **PPI** network of choice.

6.4 EXPLANATIONS FOR KNOWN CANCER GENES

With EMOGI being validated on several different sets of cancer genes, another way to establish trust in the method is through investigating important features for the classification of individual genes. As seen in Section 4.6.1, the important features for the classification of individual data points can differ with non-linear **ML** models.
To further validate the predictions made by EMOGI, feature and network contributions of well-known cancer driver genes were compared with literature knowledge about them. To explain the decisions made by the EMOGI model, **LRP** [158, 173] was employed on a trained model for a certain gene of interest. As introduced in Section 4.6.3, **LRP** feeds the output probability of a gene being involved with cancer malignancies back into the model and propagates in back to the input space. Applied to EMOGI, **LRP** identifies not only the most contributing features but also the interaction partners of those gene products that were most important for the classification of the gene of interest (see Section 5.8 for an explanation of how the **LRP** framework was adapted to work with **GCNs**).
Cancer genes were selected from the predictions that are known to be either highly mutated, amplified or deleted, differentially methylated in promoter regions or differently expressed in several different cancers. Their **LRP** values were inspected for all four *omics* across the 16 cancer types, along with their most important interaction

partners from the **CPDB PPI** network.
First, the tumor suppressor gene *APC* which has been described in the literature as highly mutated in colorectal cancer and shown to activate the Wnt signal transduction pathway in nascent intestinal tumor cells [223] was analyzed. EMOGI correctly identifies mutation rates in colon and rectal tissues as the most relevant features for classification.

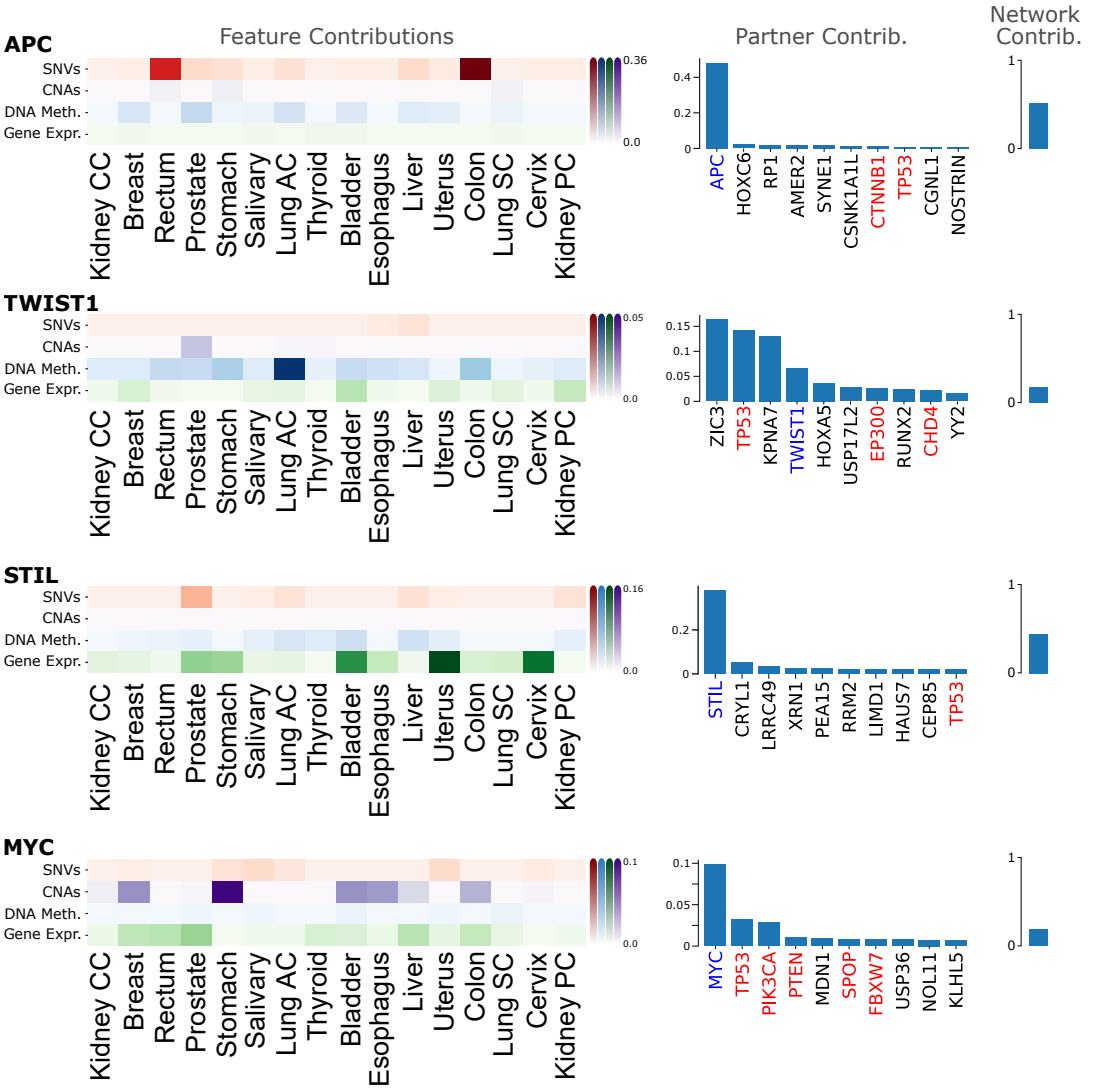


Figure 6.9: **Feature importance for known cancer genes.** The feature and network contributions for four well-known cancer genes were analyzed using **LRP**. The first column depicts the multi-omics features for the 16 different cancer types. Colors correspond to the data type (red for **SNVs**, purple for **CNAs**, blue for DNA methylation and green for gene expression). Darker colors correspond to higher importance. The second column depicts the 10 most important interaction partners of the gene and the third column shows the percentage of contributions from the **PPI** network, relative to the importance of the multi-omics features. Interactions with known cancer genes are marked in red and the contribution of the gene itself is marked in blue.

Next, the transcription regulator *TWIST1*, which plays essential roles in tumor initiation, invasion and metastasis in a variety of cancers [224] was investigated using **LRP**. *TWIST1* promoter hypermethylation, one of the most important factors in the epigenetic reprogramming of *TWIST1*, has been identified in cancers of different origins and suggested to be a useful biomarker for screening colorectal tumors [225]. EMOGI correctly identifies DNA methylation in lung cancer and colorectal cancer,

followed by kidney and thyroid cancers, as the most important features for classification of *TWIST1* as cancer gene. Furthermore, **CNAs** in prostate adenocarcinoma are identified as a potential novel molecular mechanism by which *TWIST1* contributes to the cancer phenotype for which evidence was already found previously [226, 227].
Third, explanations for *STIL* were investigated; a gene that was reported to be highly overexpressed in multiple cancer types [228]. EMOGI identified gene expression in uterine and cervical but also other cancer types as important contributors to the classification of *STIL* as cancer gene.
Lastly, the famous oncogene *MYC* was examined, a gene which is often amplified [20, 60] and overexpressed [37, 229] across cancer types. Accordingly, **CNAs** across many cancer types are found to be most important for the classification of *MYC* as cancer gene. Additionally, gene expression is the second-most important data type for the classification of *MYC*.
LRP explanations were computed for four additional **KCGs** and are depicted in the appendix as Figure A.8. They include *E2F1*, a key regulator of DNA repair and known to be abnormally expressed in cancer cells [59]. Here, EMOGI identifies gene expression across different cancers as the most important feature, together with high mutation rate in rectal cancer, in line with a previous study showing that somatic mutations that disrupt microRNA target sites in the E2F1 mRNA lead to a mechanism of oncogene activation in colorectal cancer [230]. Additionally, *KRAS* was examined, a highly mutated **KCG** across cancer types [231]. Finally, to complement the **LRP** analysis, two **KCGs** were investigated that are mostly classified because of their interactome. These include *CREBBP*, where especially HOX **TFs** were reported to be important interaction partners of *CREBBP* [232–234] and *NRAS*, a RAS signaling oncogene [235] that is rarely mutated [236] but can promote tumor growth substantially [231, 237].

As introduced in Section 5.6, **CV** allows measuring the uncertainty of predictions empirically by computing well-known measures, such as the standard deviation (Std, denoted σ). This also holds true for the **LRP** explanations which are computed for each **CV** EMOGI model separately (see Section 5.8 for an explanation). A striking correlation between the certainty of the explanations and the EMOGI score (average output probability across **CV** models) can be observed (spearman correlation $R = 0.84$, p-value $< 2.2e^{-16}$). The correlation is depicted as contour plot in the appendix as Figure A.9 and describes an arc where highly predicted and rejected genes are associated with low uncertainty but intermediate predictions have uncertain explanations.

Not only the omics features, but also the gene interaction partners in the **PPI** network might contribute substantially to the classification of cancer genes. For example, EMOGI predicts that the most important interaction partners of the **TSG** *RB1* are the *E2F1* transcription factor, known to be regulated by RB1, and the histone deacetylase *HDAC1* (depicted in Figure 6.10). This is in line with numerous previous studies which have reported that the RB1/E2F pathway regulates cell cycle progression, apoptosis and DNA repair and has been found to be disrupted in virtually all cancers [238]. Accordingly, **LRP** identifies *E2F1* as the most contributing interaction

partner of *RB1* and vice versa. Furthermore, normal expression of histone deacytelases is disrupted in various cancers and they have been reported to play crucial roles in the activation and repression of cancer genes [239]. Overexpression of *HDAC1* in particular is associated with poor patient outcomes [239]. Furthermore, *RB1* is known to recruit histone deacetylases to repress transcription of *E2F*-regulated genes [240], which would explain why EMOGI identified these strong connections between the three genes.

In addition, the SWI/SNF chromatin remodeling complex can be reconstructed in the same way. The complex was previously connected to different cancer types in a pan-cancer computational analysis [41] and also experimentally shown to be implicated in the diseases [241]. In particular, *ARID1A* was shown to be significantly mutated in bladder and uterine cancers [242] and *ARID1B* was reported to be mutated in brain cancer (juvenile neuroblastoma) [243].

And lastly, the PIK3 signaling pathway can be partly recovered and **PPIs** between

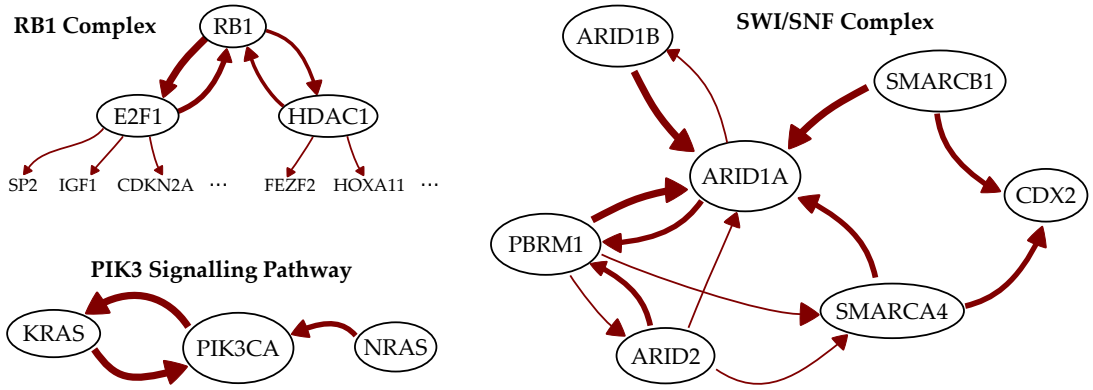


Figure 6.10: **Three cancer-related protein complexes or pathways identified through manual inspection.** The networks were collected by inspecting the most important interaction partners for genes located in important pathways or complexes, especially those identified in [41]. For each pathway/complex, only genes with highly contributing interactions to other genes of the module are depicted. Interaction strength corresponds to edge thickness.

KRAS, *PIK3CA* and *NRAS* are highly important for the classification of these genes as cancer-related genes. PIK3/AKT/mTOR signaling is an important pathway for the regulation of proliferation, apoptosis and angiogenesis (the process of growing blood vessels to assure nutrient transport to the tissue) and thus often implicated with cancer [41, 244].

Although a manual inspection of the most important molecular and network features for all predicted genes is unfeasible, the relative contribution of the network versus the omics features seems to considerably vary from gene to gene, with cancer genes such as *NRAS* and *CREBBP* being more driven by the interactome, and genes such as *KRAS* and *APC* being driven more by the genetic features of the genes.

The fact that EMOGI classifies well-known cancer genes because of previously identified alterations is further evidence that its predictions are trustworthy and that it learns to distinguish cancer from non-cancer genes through different and heteroge-

neous molecular readouts, such as genomic, transcriptomic, epigenetic and interaction factors.

6.5 THE IMPACT OF DIFFERENT OMICS LEVELS

With EMOGI being validated on several different sets of cancer genes and it being able to reproduce literature knowledge of several well-studied cancer genes, it is interesting to investigate which omics levels are most important for the identification of cancer genes on a more global scale.
To that end, the model was trained using only subsets of the four omics information. As mentioned already above, the **CPDB PPI** network was used because it yielded the highest performance for EMOGI and DeepWalk (with or without additional multi-omics features, Figure 6.7 & Figure 6.8). **AUPRC** values were computed on both the **KCGs** and **CCGs** from the **NCG** separately. While the **KCGs** were used for training and testing (see Section 5.2.6), the **CCGs** are not. As depicted in Figure 6.11, the performance increases significantly with the integration of more *omics* types for both gene sets, denoted by the arrows and associated p-values from a two-sided t-test. In particular for the **KCGs**, using only a subset of the four *omics* decreased EMOGI's performance in almost all cases. Figure A.10 depicts the pairwise statistical significance when increasing the amount of data in more detail for the *omics* levels. A statistically significant increase in performance for most of the cases was observed. An exception is the **SNV**-only model which achieves a high performance on both, **KCGs** and **CCGs**, exemplifying that mutation rates are the most important single *omics* type for cancer gene classification, as expected. However, it seems that **CNA** information boosts the performance of all models greatly when it is not the only data type presented to EMOGI. This observation is in line with the sparse nature of **CNAs** in patient samples (see Figure A.2 and Figure 5.2) and was observed previously [20]. Furthermore, gene expression and DNA methylation data appear to be partly redundant, as is expected given that highly methylated promoter regions of genes reduce their expression [61].
Interestingly, the identification of **KCGs** appears to benefit more from the heterogeneous data sets, compared to the **CCGs** because the difference between the full multi-omics setup and the reduced combinations of features is high and the increase compared to the random performance is much higher.

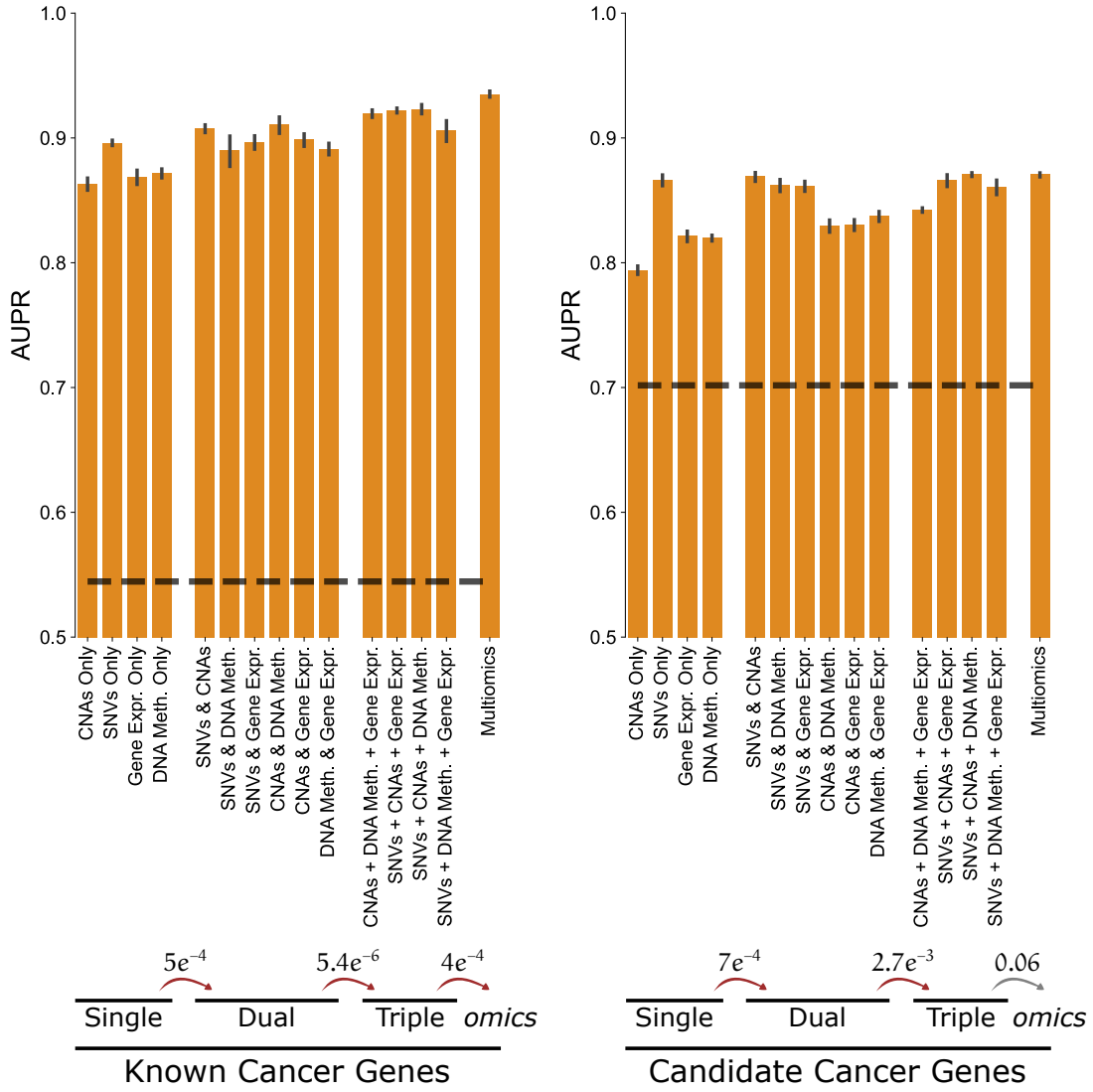


Figure 6.11: **Recovery of cancer gene sets from the NCG using only subsets of *omics* data for EMOGI training. AUPRC** values were computed for EMOGI models trained on subsets of *omics* levels using either **KCGs** or **CCGs** as reference cancer gene set on the **CPDB PPI** network. Models are grouped according to using one data type (single), two types of *omics* levels (dual), three types of *omics* (triple) or all four (multi-omics). The arrows and numbers above denote significance p-values from a two-sided t-test between the **AUPRC** of models using fewer *omics* levels and EMOGI models using more data. Gray arrows denote no significance. The black dotted line denotes the random performance for both gene sets. Error bars denote standard deviation across the different folds of the cross-validation.

For the **CCGs** EMOGI's performance is robust when using only two data types as long as one of them is the **SNV** mutation rate. This is comprehensible, given that the **NCG** candidate cancer genes represent a gene set based on elevated mutation rates in tumor samples across cancer types [194].

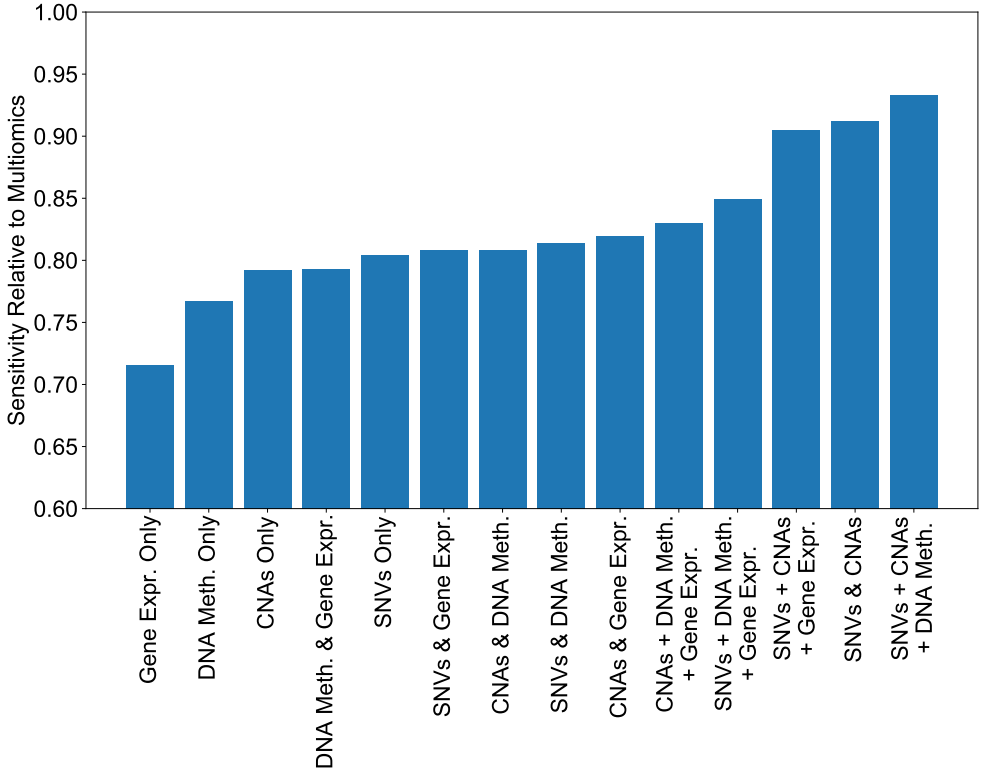


Figure 6.12: **Sensitivity for different *omics* subtypes, relative to the multi-omics setting on the CPDB PPI network.** The EMOGI models using subsets of the multi-omics features were compared in their capacity to recover **KCGs**. Depicted is the fraction of **KCGs** contained in the predictions (sensitivity) for each of the models. The cutoffs were determined based on the intersection between precision and recall. Bars represent the fraction of sensitivity for each model with respect to the full multi-omics setting and are ranked according to sensitivity. A value of e.g. 0.85 means that EMOGI trained on that subset of features was able to achieve 85% sensitivity compared to the full model in the recovery of **KCGs**.

As additional confirmation to the benefit of using multi-omics node features, the percentage of detected known cancer genes for EMOGI trained on a subset of the *omics* types was compared to the full multi-omics setting (Figure 6.12). All subsets of the full data only recover a subset of what the full EMOGI model predicts. Gene expression information appears to be least predictive on its own, resulting in 72% recovered known cancer genes while the combination of **SNV**, **CNA** and DNA methylation appears most informative with almost 93% recovery rate. However, while a lower recovery rate means that less known cancer genes were successfully predicted, it does not mean that the models necessarily predict the same genes. A further comparison of the overlap between predictions is given in Figure A.11 and highlights that most of the predictions are shared between the models, which is reassuring. Furthermore, each data type brings about several unique predictions but those are lower for the multi-omics model compared to the models trained on only one *omics* level.
The analysis of EMOGI's performance on a subset of *omics* data shows empirically

that the integration of orthogonal data types is in part responsible for the high performance of EMOGI compared to other tools and approaches.

6.6 PERTURBATION EXPERIMENTS

In an additional validation, it was tested if the **PPI** network structure or the multi-omics features were most informative for the graph-based data integration method presented in this thesis. In Section 6.4, we already saw that some genes are almost entirely predicted because of their neighborhood in the **PPI** graph while others are classified as cancer-related genes because of their genomic, epigenetic or transcriptomic alterations in cancer.
To investigate the benefit of network or multi-omics features, several perturbation experiments were performed in which either the network edges or the feature vectors of individual genes or both at the same time were perturbed. The decrease in performance compared to the original model was used as a measure for the importance of the data but also as a measure of robustness of the model. A model that still performs well under conditions of slight or moderate perturbations is more trustworthy. In addition, it would imply that evidence for a gene to be associated with cancer can come from interaction partners or features alike.

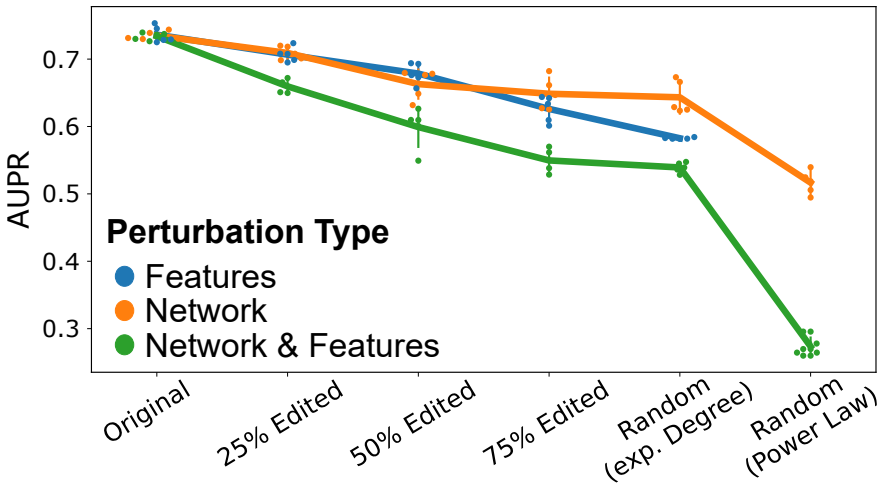


Figure 6.13: **Feature and network perturbation results.** The full EMOGI model was perturbed by either shuffling feature vectors of genes, shuffling **PPIs** or both. EMOGI models were trained on the perturbed data using 5-fold **CV** and **AUPRC** values were computed for the perturbed models. Error bars denote standard deviation, the raw performances for the perturbed models are shown as dots and the line follows the average performance across **CV** runs.

Therefore, a systematic perturbation of both node features and network interactions was conducted. For the features, two genes were randomly selected and their entire feature vectors were exchanged. This process was then repeated for either 25%, 50% or 75% of the nodes. A complete randomization was achieved by generating random feature vectors of length 64.

The network perturbations, on the other hand, were generated using a double-edge swap. Here, two edges are randomly selected (e.g. u — v and x — y) and rewired such that one obtains two new edges u — x and v — y. This procedure is also repeated until either 25%, 50% or 75% of the edges are swapped. This procedure preserves the node degree of all nodes because all four nodes receive and lose a connection [245]. It was shown in the past that great care is needed when predicting cancer genes or modules from **PPI** data because cancer genes tend to have significantly higher node degrees compared to other genes [41, 68] due to study bias [217]. To completely randomize the **PPI** graph, two scenarios were considered which either preserve the node degree or not. To evaluate EMOGI on a completely random graph that preserves the node degree, a random network was generated using the algorithm by Miller and Hagberg [246]. To also randomize the node degree of genes, a second random network was generated that does not preserve the node degree but which loosely resembles biological networks (the node degree follows a power law distribution). For that, the algorithm from Holme and Kim was used [247].
EMOGI was trained on all of the combinations of perturbed networks, features or both and evaluated in comparison to the original multi-omics, non-perturbed model. Figure 6.13 depicts the performance of all such models in a compact way.
Perturbing only one data type per time, either the *omics* features or the network's edges, significantly reduced EMOGI's AUPRC values at each step (according to a t-test and a significance level of 0.05, not shown), except for the transition from 50% to 75% and from 75% to random for the network perturbation. However, EMOGI using even significantly corrupted data was performing much better than a random classifier—which would only yield an **AUPRC** value of 0.27 - especially when only perturbing one type of data. This shows a high robustness to small perturbations and noise in both, multi-omics data and **PPIs** and is recomforting because a substantial amount of noise can be expected to be present in any **NGS** data generation process [54, 82, 86, 93] (introduced in Chapter 3).
Jointly perturbing both data types still yielded **AUPRC** values of about 55% when the node degree distribution of the network was preserved, indicating that the topological features of the **PPI** network, such as node degree, can already distinguish, to a certain extent, between cancer and non-cancer genes. Randomization of all node features and network edges without preserving node degree significantly reduced EMOGI's **AUPRC** down to about 27%, which corresponds to the performance of a random classifier (depicted in Figure 6.7).

These experiments show that both network and *omics* features are important and non-redundant in ensuring the model's accuracy, justifying the use of a complex model that incorporates diverse data representations for the task of predicting cancer-related genes.

6.7 THE IMPACT OF A PAN-CANCER ANALYSIS

In a final validation of EMOGI, it was investigated if the model benefits from the pan-cancer approach, as hypothesized theoretically in Section 2.2.2 or if it is even better suited in a cancer type specific setting.

To recapitulate briefly, it was observed that while many cancer genes are specific to a

Breast Cancer

Sensitivity Analysis

Method	BRCA Training Set	BRCA Test Set	DriverDB Breast Cancer Genes
Pan-Cancer	68%	46%	31%
BRCA Averaged	54%	39%	22%
BRCA Patient-Wise	68%	46%	20%

Thyroid Cancer

Sensitivity Analysis

Method	THCA Training Set	THCA Test Set	DriverDB Thyroid Cancer Genes
Pan-Cancer	85%	59%	46%
THCA Averaged	6%	0%	3%
THCA Patient-Wise	3%	6%	1%

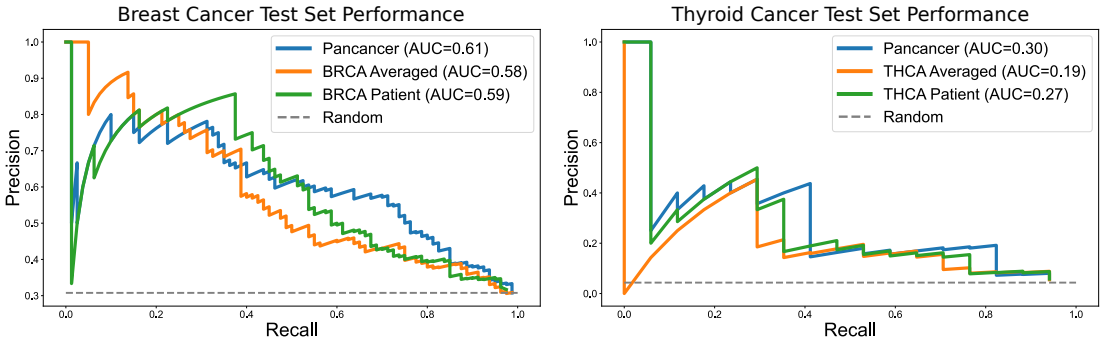


Figure 6.14: **Performance of cancer type specific EMOGI models.** The two cancer type specific EMOGI models were compared to the pan-cancer model. Tables on the top depict sensitivity of patient-wise and averaged EMOGI models in comparison to the full pan-cancer model on the cancer-specific training and test sets, as well as an independent set of literature-mined cancer-specific driver genes from DriverDB [191]. The cutoffs were chosen based on the intersection between precision and recall. At the bottom, precision-recall curves compare the performance of the three models (pan-cancer, patient-wise and averaged) for breast and thyroid cancer on the breast or thyroid cancer test sets.

type of cancer, several genes are recurrently altered across different types of cancers [30, 41]. This observation holds true not only in light of **SNVs** and **CNAs** where the long-tail phenomenon (stating that some genes are frequently mutated while a majority of cancer driver genes are only rarely mutated) has long been recognized, but also for epigenetic and transcriptomic data. Cancer cells often acquire stem cell-like features and similar genes are epigenetically altered or differentially expressed across cancer types [248]. Hence, previous studies have argued that using multiple cancer types simultaneously gives more statistical power to an algorithm to detect rarely altered cancer genes [30, 41].

To assess whether training EMOGI on pan-cancer data has an advantage over a model trained on a single cancer type, cancer type-specific EMOGI models were created for two cancers: breast cancer (BRCA), a well-studied cancer type where many marker genes are known, and thyroid cancer (THCA), which encompasses a cohort of similar size in **TCGA** (see Table A.1), but is less studied in terms of marker genes compared to BRCA.

For both cancer types, specific training, validation and test sets were collected using the COSMIC cancer gene census [195] which contains, for every gene, the cancer types where the gene is most likely a driver gene. As for the pan-cancer model, the set of positives was enriched using the DigSEE database [199], resulting in 496 breast

cancer genes and only 65 thyroid cancer genes. 2187 negatives (non-cancer genes) were collected as for the pan-cancer model. To obtain good **HPs**, a grid search was used as depicted in Figure 5.8 and explained in Section 4.2.3, although fewer parameter ranges were tried because reasonable settings of the pan-cancer model were partly already discovered and could be reused.

To investigate the behavior of EMOGI on a cancer-specific data set, EMOGI was trained on the **CPDB PPI** network averaging the values of each *omic* level across all patient samples of a certain cancer type, similarly to the pan-cancer setting. These models as referred to as *averaged cancer-specific* models, namely *averaged BRCA* model and *averaged THCA* model. In this setting, each gene is associated with a four-dimensional feature vector that encompasses average mutation rates, copy number changes, differential DNA methylation at the promoter and differential expression of the gene.

Second, for both cancer types a model was created where the *omics* values across samples are not averaged. Instead, an additional rank was added to the multi-omics feature matrix X, making it a tensor of rank 3 (introduced in Section 5.5). The added rank corresponds to the patient samples, in order to train EMOGI directly on patient-specific *omics* features. The input tensor $X \in \mathbb{R}^{N \times S \times 4}$ contains N genes as rows, S samples as columns and 4 *omics* levels as channels, similar to a digital representation of a rgb color image. These models are referred to as *patient-wise* models, namely *patient-wise BRCA* and *patient-wise THCA* model.

Next, the cancer-specific models were systematically compared to the pan-cancer EMOGI model, depicted in Figure 6.14. Due to the collection of cancer-specific training, validation and test sets, a direct comparison on either test set is inherently unfair. Because the pan-cancer model is conceptually expected to also recover cancer-specific genes, all three models were compared on the cancer-specific test sets.

The pan-cancer model achieves higher sensitivity on an independent cancer gene set from DriverDB [191] and higher **AUPRC** compared to both the *averaged cancer-specific* and *patient-wise* models on the respective cancer-specific test set. The advantage of a pan-cancer model is expected to be less pronounced, given the high number of known breast cancer genes used for training the cancer-specific models. And indeed, sensitivity on both, the training and test set is equivalent for the patient-wise and pan-cancer model for breast cancer (Figure 6.14 top left). Nonetheless, even in this well-studied cancer type, the pan-cancer model achieves higher **AUPRC** values, especially in the high-recall regime (the right part of the PR curve). Interestingly, for both cancer types a *patient-wise* model, which captures patient variability during training, achieved a better performance than the *averaged cancer-specific* models.

The advantage of pan-cancer analysis becomes strikingly more evident for thyroid cancer, where fewer marker genes are known and the cancer-specific models struggle to achieve a good performance. Here, the pan-cancer model achieves a 10-fold higher sensitivity on an independent set of thyroid cancer genes from DriverDB [191] while the thyroid-specific models appear barely trainable, given the few marker genes for that disease.

Further insights can once more be gained from explaining EMOGI's decisions. Feature explanations were computed using the pan-cancer model for three breast cancer genes, *PRDM2*, *SIRPA* and *POLG*, all of which are contained in the COSMIC CGC but are missed by both breast cancer models, yet predicted by the pan-cancer model. This

is because their alterations in cancer types other than breast cancers contributed to their correct classification, pointing out once more the benefits of using a pan-cancer approach (Figure 6.15). While *PRDM2* and *SIRPA* show **CNAs** and DNA methylation in breast cancer as the most prominent feature for their classification, *POLG* is mainly predicted because of its elevated mutation rates and **CNAs** in other cancer types. Interestingly, *SIRPA* is mostly predicted because of its interaction partners while *POLG* appears to be more often mutated across cancer types.

6.8 SUMMARY

EMOGI is a pan-cancer graph integration method which benefits from complementary information represented by a feature matrix and a network (shown in Section 6.1) and which can faithfully predict cancer driver genes from heterogeneous molecular data sets. We have seen that it outperforms various other algorithmic approaches for the prediction of cancer genes in a variety of experimentally derived, manually curated or computationally predicted cancer gene sets (shown in Section 6.2). Various perturbations have shown that the complexity of the non-linear classifier as well as the integration of heterogeneous data types is justified because removing or perturbing any of those results in a lower performance (shown in Section 6.5 and Section 6.6). Additionally, the use of a pan-cancer approach outperforms cancer-specific models. Interestingly, a patient-wise EMOGI model achieves stable performance on a breast cancer cohort, indicating that this modification of the algorithm could be used to stratify patients in the future. Finally, the model appears to base its predictions on molecular mechanisms that have been discovered previously and that are reasonable for a variety of well-studied cancer genes (shown in Section 6.4).
Taken together, the results from this chapter show convincingly that EMOGI is a powerful, trustworthy and robust algorithm for the prediction of cancer-related genes.

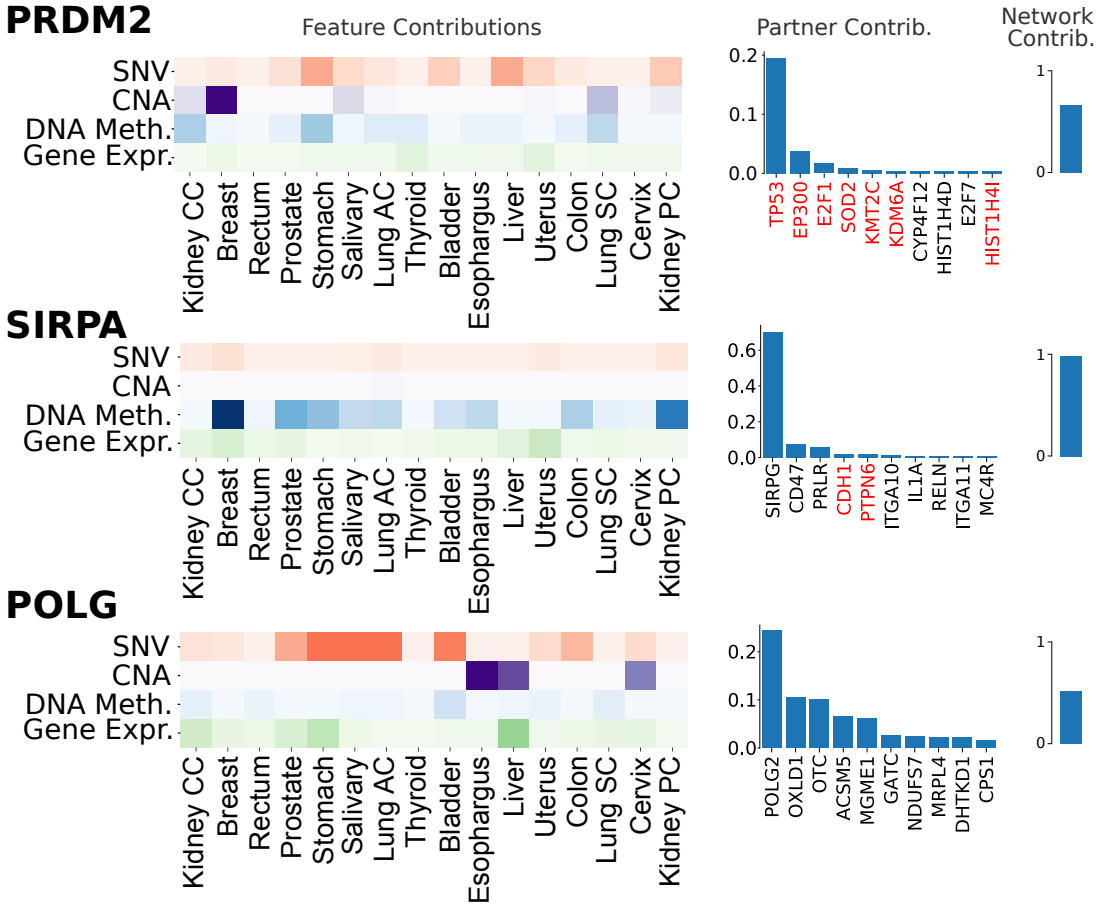


Figure 6.15: **Explanations from pan-cancer model for selected breast cancer genes.** The first column depicts the multi-omics features for the 16 different cancer types. Colors correspond to the data type (red for **SNVs**, purple for **CNAs**, blue for DNA methylation and green for gene expression). Darker colors correspond to higher importance. The second column depicts the 10 most important interaction partners of the gene and the third column shows the percentage of contribution from the **PPI** network, relative to the importance of the multi-omics features. Interactions with known cancer genes are marked in red.

7

PREDICTING CANCER-ASSOCIATED GENES WITH EMOGI

After a thorough validation of EMOGI, we have seen that it is robust across manually curated, computationally predicted and experimentally derived cancer gene sets. On the test set, EMOGI outperforms various cancer-specific and general **ML** and graph algorithms by 3% – 37% **AUPRC** values and the most important features for the classification of important and well-known cancer drivers are in accordance with literature knowledge.
Next, we set out to examine novel predictions of the algorithm and see if they are reasonable and promising new predictions. We then take the interpretability framework to the next level, truly exploiting the multi-omics integrative approach, and attempt to derive classes of cancer genes based on which data sets and cancer types drive the classification of genes. Our results bring us closer to a more fine-grained definition of what a cancer driver gene can look like, and makes us appreciate the vastly different ways in which a gene can influence cancer cell growth.

7.1 NEWLY PREDICTED CANCER GENES

The first step to gain new insights about cancer diseases from EMOGI is by examining novel predictions. Those **Newly Predicted Cancer Genes (NPCGs)** are predicted with a very high probability but are not previously annotated as cancer genes in common databases. Such a list of novel candidates can afterwards be evaluated more closely.

7.1.1 *Deriving Top Predictions From Multiple Models*

In order to compile a high-confidence list of **NPCGs** without biasing the results towards a specific **PPI** network, top predictions from all six EMOGI models were aggregated. In more detail, the top 100 predictions from all six models were collected and those genes that were not previously annotated as cancer genes were extracted. This was done by removing all genes that were part of the positive set (see Section 5.2.6 for details). The list of **NPCGs** was ranked according to the number of EMOGI models that the gene was among the top 100 predictions. This yielded a list of 165 **NPCGs** that were then used for further analysis (a complete list of all **NPCGs** can be found in Table A.2).

7.1.2 *Novel Predictions Interact with Known Cancer Drivers*

In a first attempt to characterize the novel predictions beyond their high EMOGI score, interactions of **NPCGs** with **KCGs** was assessed. The rationale behind that being that cancer driver genes are likely to be located in the same pathways or protein complexes as **KCGs** as explained in Section 2.1.3 but do not have to be significantly

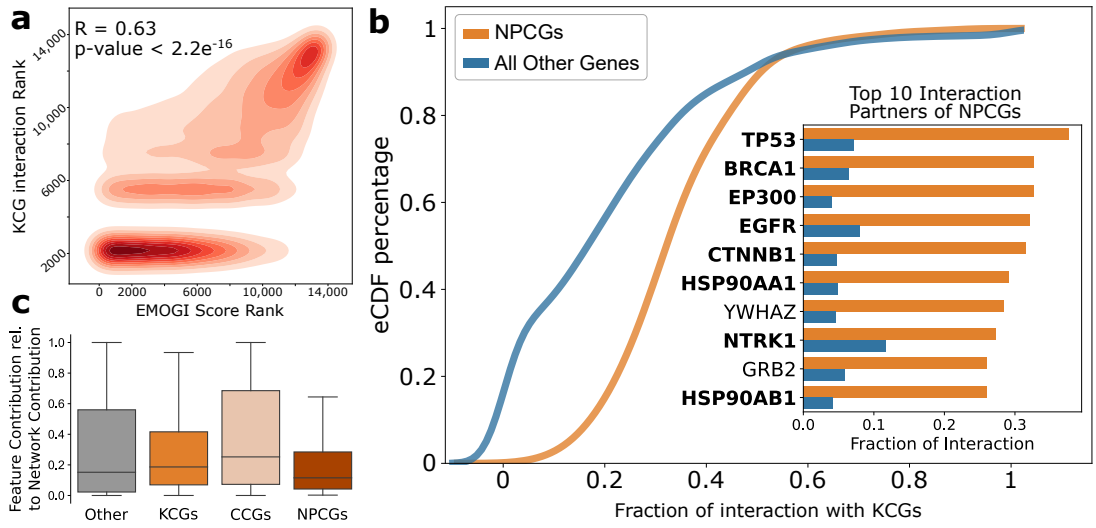


Figure 7.1: **Newly Predicted Cancer Genes (NPCGs) interact with known cancer genes. a** Rank correlation plot between the EMOGI score for each gene (output probability, x-axis) and the number of interactions of that gene with known cancer genes (y-axis) in the **CPDB PPI** network. Genes with high predicted probability of being cancer genes tend to interact with more **KCGs**. **b** Interactions with **KCGs** for **NPCGs** specifically, visualized as cumulative density function of the fraction of interactions that occur with **KCGs** for both **NPCGs** and all other genes. The top ten interaction partners of **NPCGs** are also shown. Orange bars correspond, for each gene, to the fraction of interactions with **NPCGs** and blue bars corresponds to the total fractions of interaction partners from the **PPI** network. The genes depicted here are ordered by the fraction of interactions with **NPCGs**. Known cancer genes are highlighted in bold. **c** Average fraction of the contribution of the *omics* data versus the **PPI** network, as computed through the **LRP** framework for **Known Cancer Genes** (**KCGs**), **Candidate Cancer Genes** (**CCGs**), **NPCGs** and Others, where this last group refers to genes which are not present in any of the previous three sets.

altered genetically or epigenetically themselves.

Figure 7.1a shows a contour plot, visualizing the genome-wide correlation between EMOGI score — representing the probability of a gene of being a cancer gene — and the number of interactions that the gene's protein has with **KCGs** (Spearman correlation 0.63, p-value $< 2.2e^{-16}$). Genes with a high-ranked EMOGI score are top predictions. A significant correlation between the EMOGI score and the number of interactions of that gene with **KCGs** was observed. Strikingly, all of the **NPCGs** had at least one interaction with a **KCG** and **KCGs** were also significantly enriched among the top 10 interaction partners of **NPCGs**. This is depicted in Figure 7.1b, where the fraction of genes from the **NPCGs** that interact with the gene in question is depicted in orange. For example, the well-known cancer driver genes *TP53*, *BRCA1*, *EP300* or *EGFR* were among the 10 genes that most interact with **NPCGs**. This observation was not biased towards genes with a high degree in the **PPI** network, as the fraction of interactions that occur with **KCGs** is higher for the **NPCGs** compared to all other genes, depicted in the cumulative density function (Figure 7.1b).

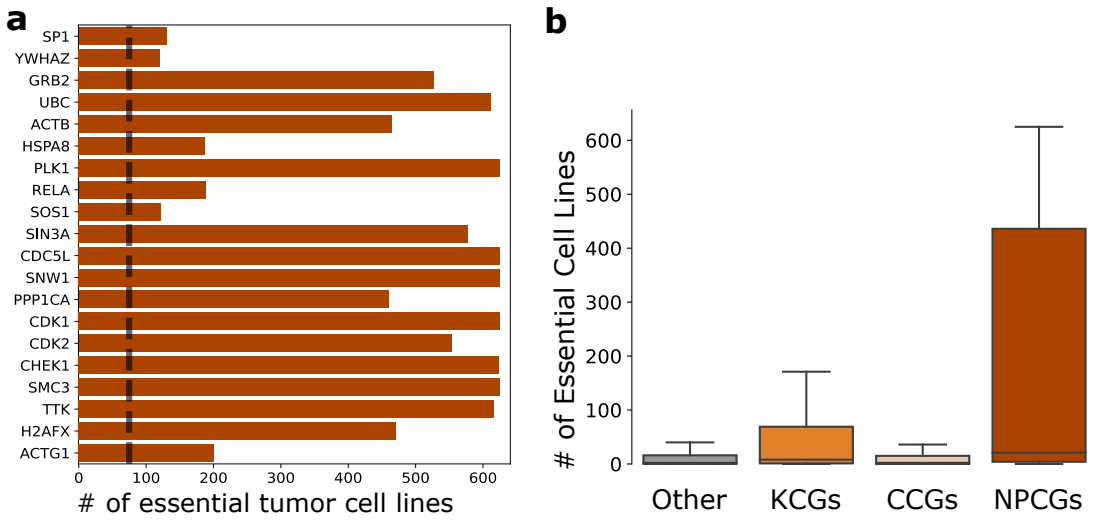


Figure 7.2: **Newly Predicted Cancer Genes (NPCGs) are essential genes in CRISPR interference (CRISPRi) loss-of-function screens. a** The top 20 **NPCGs** that have a significant negative growth effect (CERES score $\leqslant -0.5$) on tumor cell lines from the Achilles Project with the corresponding number of affected tumor cell lines. The dotted black line corresponds to the average number of affected cell lines. **NPCGs** are enriched for essential genes (Fisher's exact test, p-value $= 4.9e^{-11}$, odds ratio $= 3.1$). **b** Fraction of affected tumor cell lines for **Known Cancer Genes** (**KCGs**), **Candidate Cancer Genes** (**CCGs**), **NPCGs** and Others, where this last group refers to genes which are not present in any of the previous three sets, similar to Figure 7.1c.

This demonstrates that **NPCGs** are located in closer proximity to **KCGs** in the **PPI** network than expected by chance. This in turn is strong evidence that **NPCGs** are often located in the same pathways or complexes as the **KCGs**, making these novel predictions interesting candidates for drug targets that might not have been previously characterized.
This result was validated from another perspective, exploiting the potential of EMOGI's explainability. The **LRP** framework was applied to extract the contribution of the network versus the node features to the classification of the **NPCGs**, as explained in Section 5.8. This time, however, **LRP** was conducted for all genes in the **CPDB PPI** network, enabling comparison between different gene sets. As depicted in Figure 7.1c, the overall contribution from the multi-omics features is low in comparison to the other gene sets and hence, the contribution from the interactome (the **PPI** network) was higher for **NPCGs** than for any other gene set. Interestingly, it was even higher than for the **KCGs**, indicating that the **NPCGs** indeed represent a set of genes whose main reason for being classified as cancer genes is the interactome.

7.1.3 *Novel Predictions Are Essential in Tumor Cell Lines*

The validation of cancer genes experimentally is not straightforward. Knockout or knockdown experiments in humans are impossible for obvious reasons and systemat-

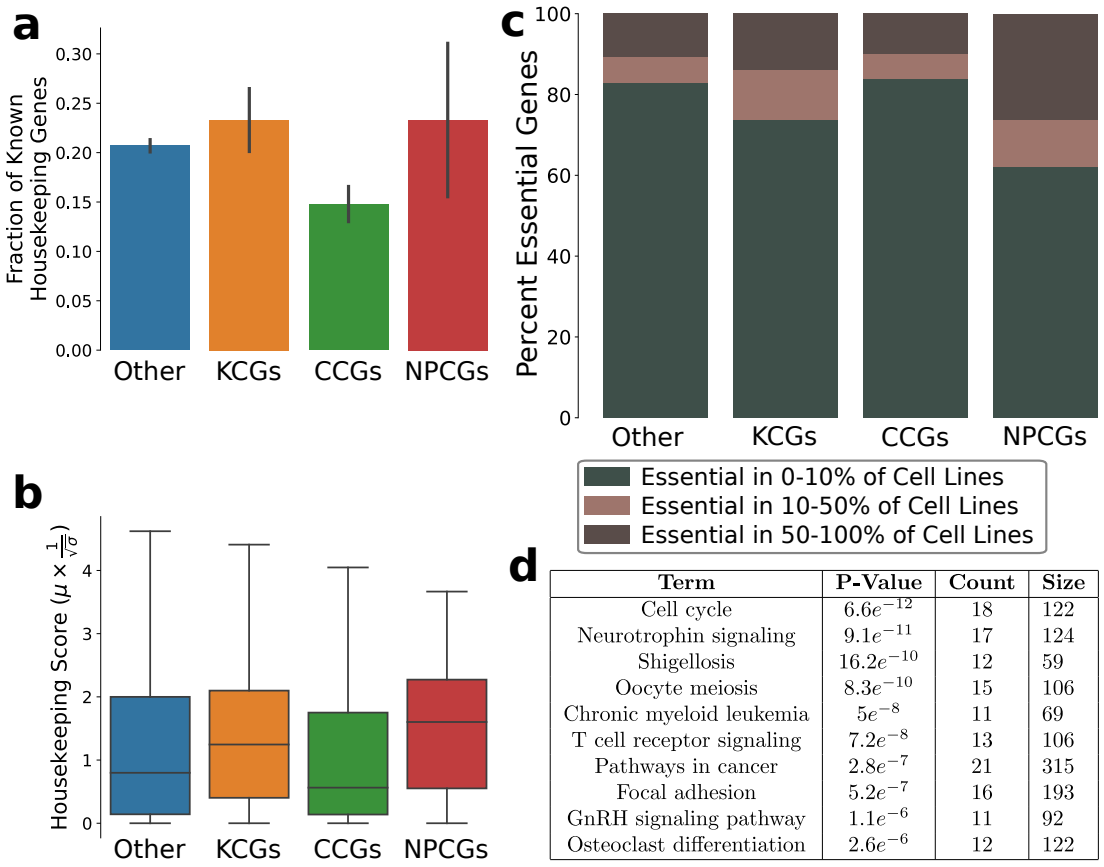
ically perturbing genes in mice not only is expensive and time consuming but would also mean an enormous sacrifice of mice. However, immortalized tumor cell lines exist and can be perturbed in high-throughput loss-of-function screens [249]. Such screens systematically knock down one gene at a time using **CRISPRi** or **RNA interference (RNAi)** systems. In the Achilles project [249], such screens were done for all genes across 625 (publicly available) tumor cell lines. The effect of the knock-down on the cell culture growth was then measured, enabling the *in vitro* validation of EMOGI predictions.

In the project, each gene is assigned a so-called *CERES* score in each cell line which summarizes the effect of that gene on cell proliferation from the loss-of-function screen [250]. A score $\leqslant -0.5$ means that the knock-down of the gene significantly reduced the culture growth, indicating a potential oncogene for the cell line). Similarly, a score $\geqslant 0.5$ means that the knock-down significantly promotes tumor cell growth and that the perturbed gene is a potential **TSG**.

However, it was observed that positive CERES scores are often due to random effects [1] [250] and were therefore not considered in the validation. To derive a single *essentiality* score per gene that indicates how important a gene is for the survival of tumor cell lines independent of the cell line, a significant effect (CERES score $\leqslant -0.5$) for more than 78 tumor cell lines was considered as a threshold. That is, if a gene negatively affects the growth of more than 78 tumor cell lines, it is defined to be essential. The threshold of 78 corresponds to the average number of cell lines affected by a gene and therefore, essential genes affect more cell lines than the average.

NPCGs were significantly enriched in essential genes (p-value$= 4.4e^{-15}$, Fisher's exact test). Among the top 20 essential NPCGs depicted in Figure 7.2a, there were genes that affected up to 600 tumor cell lines, such as the ubiquitin protein *UBC* which has been associated with **DNA** repair and apoptosis, cyclin-dependent kinase *CDK1* or the polo-like kinase *PLK1* which is associated with cell cycle and gliomas. Furthermore, **NPCGs** significantly decreased tumor cell line growth of more cell lines on average, compared to **KCGs** and **CCGs**, as depicted in Figure 7.2b. The difference is highly significant and interestingly, **NPCGs** have a stronger effect on cell line survival than even the **KCGs** (p-value $= 0.015$, fisher's exact test). Loss-of-function screens such as the data from the Achilles project have several drawbacks, however. First, the cells grow in an artificial system where they can grow outside of their normal microenvironment. An immune system is not present in cell line cultures, for instance, that heavily influences selective growth advantages of cells *in vivo* because it will detect tumor cells that cannot adapt adequately. Second, there are no such screens for normal cell lines of the same tissues. This makes it hard to tell if an effect of a perturbation will affect all cells or only the cancer cells. The latter is especially important because it could be that **NPCGs** are enriched with so-called *housekeeping genes*. These are genes required for the normal functioning of any cell in any tissue [251] and are not of special interest as potential drug targets because their disruption would not necessarily have a stronger effect on tumor cells than normal cells.

1 https://depmap.org/portal/faq/#dep_positive_ceres_score



d

Term	P-Value	Count	Size
Cell cycle	$6.6e^{-12}$	18	122
Neurotrophin signaling	$9.1e^{-11}$	17	124
Shigellosis	$16.2e^{-10}$	12	59
Oocyte meiosis	$8.3e^{-10}$	15	106
Chronic myeloid leukemia	$5e^{-8}$	11	69
T cell receptor signaling	$7.2e^{-8}$	13	106
Pathways in cancer	$2.8e^{-7}$	21	315
Focal adhesion	$5.2e^{-7}$	16	193
GnRH signaling pathway	$1.1e^{-6}$	11	92
Osteoclast differentiation	$2.6e^{-6}$	12	122

Figure 7.3: **NPCGs are not solely housekeeping genes. a** The fraction of known housekeeping genes derived by [251] present in **KCGs**, **CCGs**, **NPCGs** and Others. Error bars denote the 95% confidence interval. **b** The housekeeping score for the same four gene sets. The score is derived using **RNA-seq** data from the **Genotype-Tissue Expression (GTEx)** consortium and combines mean and variance of gene expression across tissues. **c** Fraction of essentiality of genes among the four different gene sets. Each gene is catalogued by the number of tumor cell lines for which it is essential (significantly decreases tumor cell growth). **d** Pathway enrichment analysis for **NPCGs** using the KEGG database. The top 10 enriched pathways for the 212 **NPCGs** are shown (a complete list of enriched pathways can be found in Table A.3). In the table, *Count* denotes the number of **NPCGs** included in the pathway and *size* denotes the pathway size.

The high number of **NPCGs** that are essential for the survival of tumor cell lines directly raised the question whether the novel predictions from EMOGI are mainly housekeeping genes whose alteration is lethal in any cell. Eisenberg and Levanon [251] defined a list of 3,804 housekeeping genes which were used as a validation. As depicted in Figure 7.3a, **NPCGs** contain indeed a higher fraction of housekeeping genes compared to other gene sets but this fraction was still at around 30–35%.
Housekeeping genes are expected to be expressed at constant levels across tissue types in normal cells. To complement the analysis from above, **RNA-seq** data from the **GTEx** consortium for 16 different tissues (the same tissues that were used for the

pan-cancer analysis) was used. To assess how stable the expression of a gene is across those tissues, a *housekeeping score* was defined as:

$$HK_i = \mu_i \times \frac{1}{\sqrt{\sigma_i}}$$

where μ_i denotes the average normalized **FPKM** value for gene i across **GTEx** tissues and σ_i denotes variance of the gene across tissues. The rationale behind this formulation is that genes which are highly expressed throughout tissues receive high scores while genes that are only expressed in certain tissues are penalized because they exhibit high variance.
Again, the **NPCGs** have a higher housekeeping score on average, compared to the other gene sets (Figure 7.3b), indicating once more that a fraction of **NPCGs** are housekeeping genes.
Another way to assess the issue of housekeeping genes is to assess the number of tumor cell lines affected by the genes. To that end, for each gene the percentage of cell lines it significantly affects was compared. As shown in Figure 7.3c, more than 20% of **NPCGs** significantly reduce tumor cell culture growth of more than half of the cell lines while this number is smaller for other gene sets. However, most of the genes affect less than 10% of cell lines and the fraction of those genes that significantly reduce tumor cell culture growth of some but less than half of the cell lines is highest for **NPCGs** compared to the other gene sets.
In addition, pathway analysis of the novel predictions shows that **NPCGs** are not enriched for housekeeping functions but for signaling, cell cycle, cancer pathways and development functions (see Figure 7.3d and Table A.3 for a full list of enriched KEGG pathways), indicating that many **NPCGs** most likely exhibit cell lethality specific to cancer and not to all cells. These results taken together show that EMOGI predicts essential cancer genes without having been trained on such data. While some of them are probably required for the normal functioning of cells, this is not the case for all **NPCGs**.

7.2 CLASSES OF CANCER GENES

Early cancer sequencing studies have defined cancer driver genes as:

> [a gene] whose mutations increase net cell growth under the specific microenvironmental conditions that exist in the cell in vivo [31].

We have seen by now that novel predictions by EMOGI go beyond such a classical definition of what constitutes a cancer driver gene, in line with more recent studies that underlined the importance of epigenetic and non-coding alterations (introduced in Section 2.2). Understanding EMOGI's predictions globally and extracting groups of cancer genes guided by similar molecular mechanisms is therefore the next logical step. The key idea behind that is to use the **LRP** framework for all genes in combination with unsupervised **ML** (introduced briefly in Section 4.1) in order to reveal groups of cancer genes predicted as such because of different and unique patterns in the data.

First, EMOGI's predictions were grouped based on their most important contributing multi-omics feature contributions using bi-clustering. Second, a complementary approach was employed to find modules of **PPIs** that were important for the classification of multiple genes.

7.2.1 *Bi-Clustering of Feature Contributions*

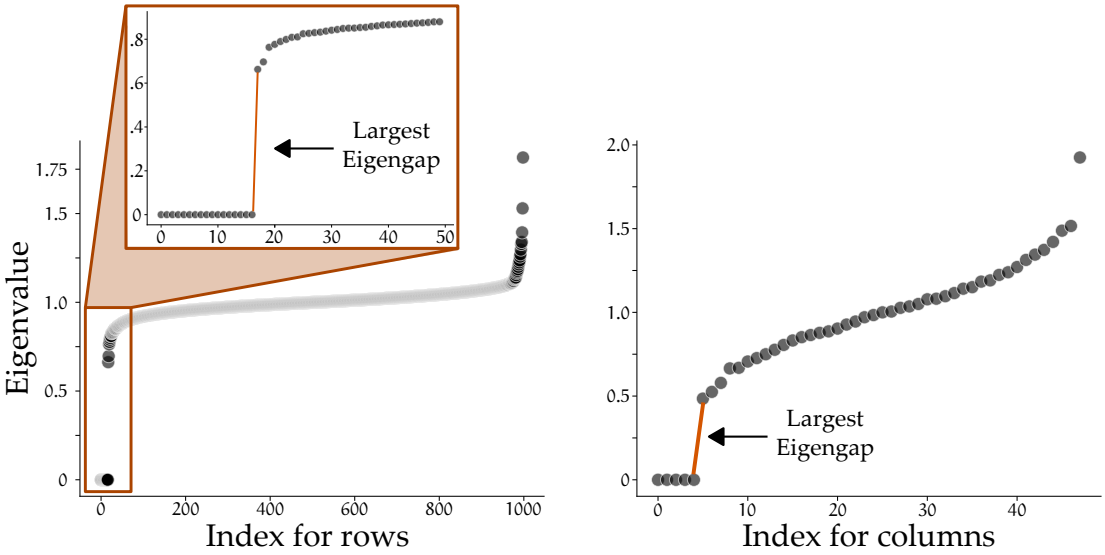


Figure 7.4: **Eigengap analysis to determine optimal cluster numbers.** X-axes depict the eigenvalues ordered by their size and the y-axes denote the value of the eigenvalues. The optimal number of clusters k is derived from the largest eigengap. For the rows, the first 50 eigenvalues are enlarged in the box.

Clustering attempts to find latent structures in a data set by grouping observations (or data points) in groups or clusters, maximizing distances between clusters and minimizing distances within clusters. While a plethora of clustering algorithms exist, two questions are fundamental for their application. First, the number of desired clusters (k) in the data is hard to define. For some applications this might be an obvious choice but mostly, there is no clear answer and additional computational methods have been developed to find good estimates. Second, a distance measure (metric) has to be defined that makes sense for the particular data set.

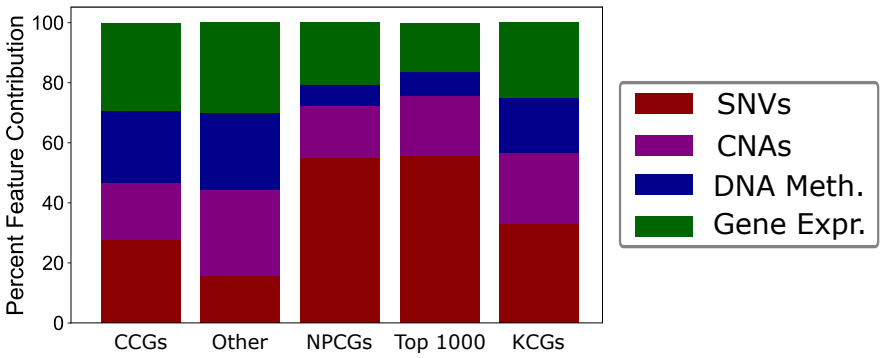


Figure 7.5: **Contribution of *omics* data across predictions for different gene sets.** Values are measured for the **CPDB PPI** network and all contributions sum to 1.

To stratify genes and data as well as cancer types at the same time, *bi-clustering* was used. This subfield of clustering attempts to reveal a checkerboard structure in any data matrix. Initially proposed for the analysis of gene expression studies with several conditions, spectral bi-clustering is a powerful bi-clustering method [252]. It works by converting a data matrix into a bipartite graph where one set of nodes denotes rows (in this case genes) in the data and the other set of nodes denotes columns (data and cancer types here). The eigenvalue decomposition of the Laplacian of that graph computes signals over that graph which do not change the feature distributions (see Section 4.5 for details on spectral graph theory and an introduction to **GSP**). Hence, the first eigenvectors can be clustered using any well-established clustering method (such as k-means) to partition the graph. This can be used to reorder the data matrix such that groups with similar rows and columns are close to one another. Initial attempts to use biclustering on the multi-omics feature contributions of all genes resulted in a partitioning between highly predicted genes and all other genes (see Figure A.13). This is reasonable because the overall amount of relevance from the **LRP** corresponds to the output probability for a gene (as explained in Section 4.6.3). Furthermore, the binary cross-entropy loss function used to train EMOGI is known to push data points to receive either 1 or 0 output probability. Hence, a fundamental difference between highly predicted genes and lowly predicted ones can be expected. To employ a bi-clustering of feature contributions nonetheless, only the top $1,000$ predictions from EMOGI were subjected to spectral bi-clustering, reducing the problem because the genes all received an output probability close to 1 and therefore **LRP** explanations are reliable (see Figure A.9). Furthermore, the number of $1,000$ reflects an upper bound to the size of current databases like the COSMIC CGC (699 genes) or oncoKB (642 genes).
For spectral clustering, the eigengap heuristic [138] has been shown to work well to find the appropriate number of clusters k, even when used in combination with **LRP** [151]. The eigengap heuristic states that an appropriate number of clusters will be found when the difference between two consecutive eigenvalues is large. An intuitive motivation for the use of eigengap heuristics is that in the ideal clustering setting, there are k different connected components in the graph. In that scenario, the first k eigenvalues will be 0 and the gap between λ_k and $\lambda_{k+1} > 0$ [253]. This also holds true for less well separated clusters [138].

In the application presented here, eigengap analysis was performed for rows and columns independently. This resulted in 20 row clusters and 5 column clusters as optimal values, as depicted in Figure 7.4.
The second open question for clustering is the choice of an appropriate metric. Here, the Euclidean distance was chosen. Negative contributions from the **LRP** were set to 0 because such negative contributions do not have a clear biological meaning. Because the **LRP** values have no specific scale or unit, they were subjected to standard scaling (denoted by Equation 5.6) prior to clustering.
Figure 7.6 depicts the result from the clustering. Interestingly, mutation rates seem to be the most important data type for the classification of top predictions by EMOGI.

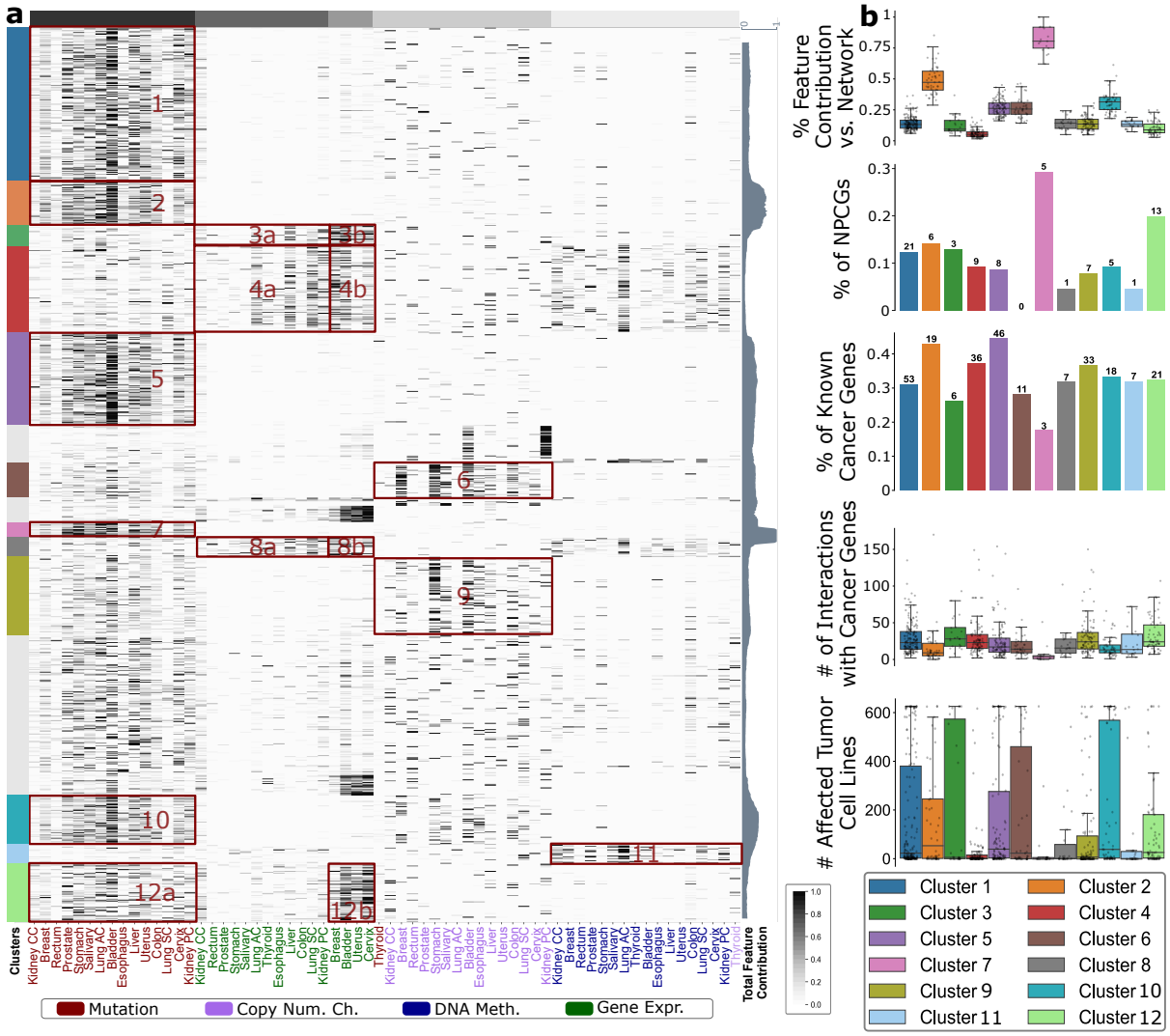


Figure 7.6: **Bi-clustering of genes and feature contributions.** Rows correspond to genes and columns correspond to *omics* types across the 16 **TCGA** cancer types. Column labels are colored according to data types. Each cell of the matrix corresponds to the **LRP** value of a certain gene for a certain *omic* level in a certain cancer type. Values have been subjected to min-max normalization. Bi-clusters correspond to the blocks defined by the partition on the left side and on the top of the matrix and representative blocks have been highlighted in red and numbered from 1 to 8. On the right side of the matrix, the cumulative contribution of all *omics* features to the gene classification, relative to the total of network and feature contribution is displayed as running average with window size 20.

This is shown by the high values in the first column cluster which includes the highlighted clusters 1, 2 and 5, 7, 10 and 12. This initial impression is further quantified in Figure 7.5 where mutation rates make up more than 50% of the feature contributions (excluding the contributions from the interactome) for the top 1000 predictions, a higher value compared to other gene sets. This is in line with our current knowledge of cancer and is expected because **KCGs** were identified as cancer genes due to their high mutation rates in various cancers and therefore included in the training data for

EMOGI. Furthermore, the high importance of mutations was already observed above, when training EMOGI on only a subset of features.
16 of the $20 \times 5 = 100$ bi-clusters were selected and highlighted in Figure 7.6a. The clusters were selected by visual inspection because they seem to represent distinct classes of cancer genes. Roughly half of the selected gene clusters corresponded to cancer type-specific mutation-driven gene predictions (cluster 1, 2, 5, 7, 10 and 12a in Figure 7.6). However, inspecting the genes included in those clusters also highlighted some important differences: The contribution of the *omics* features to the classification of genes in cluster 1, 4 and 12, for example, was much lower than the **PPI** network contribution (Figure 7.6b & Figure 7.8). In addition, cluster 12 included a high percentage of **NPCGs**, as well as the highest fraction of interactions with **KCGs** (Figure 7.6b) and several genes that are known to influence patient prognosis (Figure 7.7). It is also the only cluster showing high contributions in both, **SNVs** and gene expression for several cancer types.

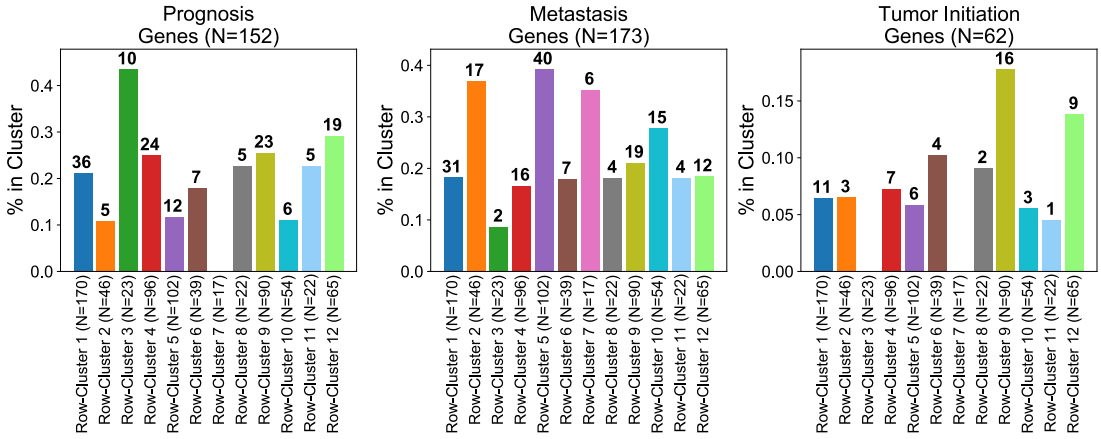


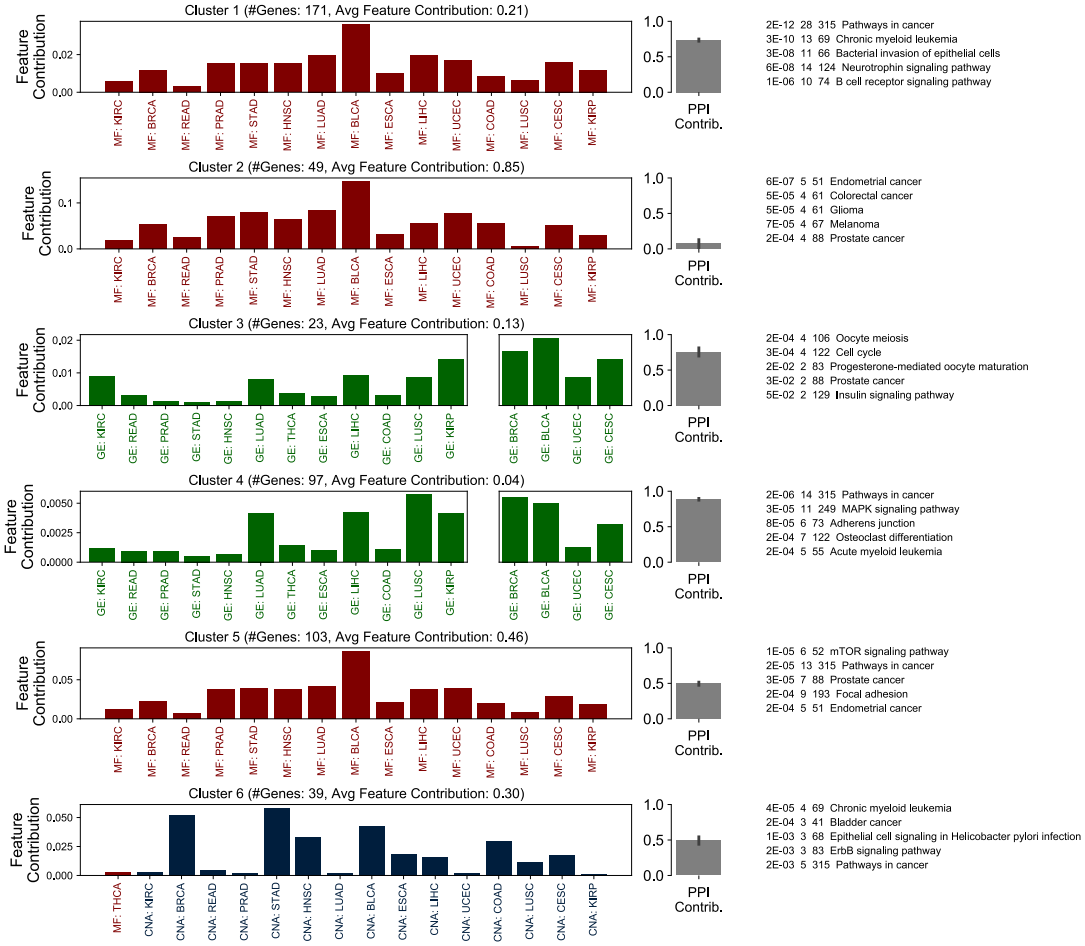
Figure 7.7: **Metastasis,prognosis and tumor initiation genes per cluster.** Prognostic cancer genes from Wee et al. [254], genes involved in metastasis from Priestley et al. [255] and genes associated with tumor initiation from CIGene [256] were collected. From the bi-clustering analysis in Figure 7.6, the percentage of prognostic, metastatic and tumor initiation-associated cancer genes contained in each bicluster was computed. The numbers above the bars denote absolute numbers. Colors match those in Figure 7.6. Note that genes can be contained in multiple gene sets, e.g. be associated to both prognosis and metastasis.

Cluster 2, on the contrary, was enriched with well-known cancer genes, such as *TP53*, *KRAS* or *PIK3CA*, where *omics* features, in particular **SNVs**, contributed more than the **PPIs** network to their classification. Consistently, this cluster was also depleted in genes interacting with **KCGs** and enriched for cancer pathways (Figure 7.6b & Figure 7.8). In addition, cluster 2 shows the highest median lethality in tumor cell lines (Figure 7.6) and was enriched for metastasis-related genes (Figure 7.7). Cluster 5, also containing mutation-driven genes, was enriched in **KCGs** participating in cancer-specific pathways and in developmental genes (Figure 7.8) which are often reactivated in cancer cells, especially to acquire migration capabilities required for metastasis [257]. Accordingly, this cluster also contained one of the highest numbers

of metastatic genes, compared to the other clusters (Figure 7.7). Interestingly, cluster 7 is a small cluster also driven by **SNVs** and highly enriched with **NPCGs**, but depleted in interactions with **KCGs**, despite the general trend of **NPCGs** to be classified because of their interactome (see Figure 7.1c).

Clusters 6 and 9 were characterized by copy number changes (Figure 7.6a) and included genes belonging to known cancer pathways (Figure 7.8), as well as genes known to be often amplified, such as *MYC* or *NRAS* in cluster 6 and cyclin-dependent kinases or the tumor initiation genes *EGFR* and *ERBB2* in cluster 9. Cluster 8 and 3, and to a lesser extent 4 and 12, where the interactome played a more important role (Figure 7.6a), are examples of groups of genes whose classification was driven by gene expression changes alone (cluster 3) or in combination with other *omics* features, such as DNA methylation (cluster 4 and 8). In particular, cluster 12 was characterized by genes exhibiting both high mutation rates and altered gene expression in a subgroup of cancer types (sub-clusters 12a and 12b).

Finally, cluster 11 included subsets of genes whose cancer classification was mainly driven by aberrant DNA methylation, among them *FOX* **TFs**, the DNA methyltransferase *DNMT3L* and the *RUNX1* **TF** that was previously reported to be differentially methylated in cancer [258, 259]. Pathway analysis shows enrichment for immune-related genes for this cluster (Figure 7.8).



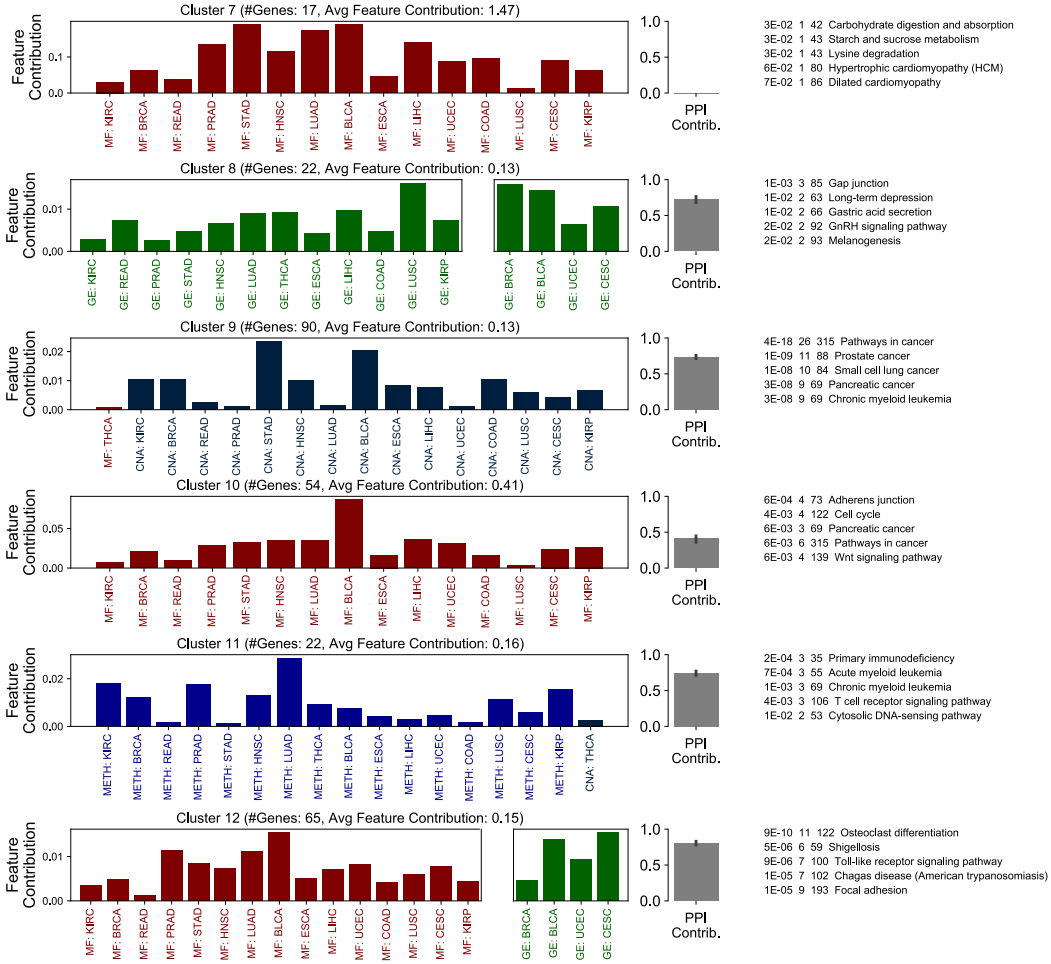


Figure 7.8: **Statistics on the biclustering analysis (continued from the previous page)** Bar plots (left column) show average *omics* feature contributions to the classification of the genes in each cluster. The middle column shows the amount of *omics* features relative to the network contribution for that cluster, and error bars denote the standard deviation across genes in the cluster. The right column shows enriched KEGG pathways in each cluster with corresponding p-values, number of cluster genes belonging to specific pathways and the total number of genes in the pathway. Split clusters are marked by a gap between the two clusters (e.g. Biclusters 3a and 3b).

7.2.2 *Summary*

All in all, the bi-clustering analysis reveals distinct classes of cancer genes. It distinguishes between interactome-driven (mainly represented in clusters 1, 4 and 12), **SNV**-driven (clusters 5, 7 and 10), methylation-driven (cluster 11), expression-driven cancer genes (clusters 3 and 8) and cancer genes being driven by **CNAs** (clusters 6 and 9). But not only the data type but also the concepts represented by the clusters are different. The concept of transcriptional addiction, for instance, hypothesizes that tumor cells often rely on individual genes to maintain their altered metabolism and that these silent players are promising drug targets. The concept exists for some time now but identifying such regulators or **TFs** from molecular data sets is difficult be-

cause gene expression alone is noisy and might not show significant differences in tumor samples. However, cluster 12 exhibits characteristics fitting very well into the concept and the genes in the cluster might be interesting to investigate further.
In addition, the bi-clustering also highlighted some cancer type specific patterns (Figure 7.8), indicating that carcinogenesis in different tissues might be triggered by different and complementary molecular mechanisms.
While the clustering procedure is highly informative to distinguish classes of cancer genes and helps to not only complement our definition of what a cancer gene might look like but also to find potential new drug targets, it only uses the multi-omics feature contribution for the clustering. But as we have seen throughout this thesis, **PPI** interactions are a vital component to the classification and **LRP** gives us scores for individual **PPIs** (as introduced in Section 5.8). Therefore, we will look at those parts of the **PPI** network that EMOGI uses most for the classification of genes in the next step to find modules and **PPI** that are generally important for tumor formation and maintenance.

7.3 MODULES OF IMPORTANT INTERACTIONS

Cancer network modules, connecting functionally related genes, help to further enhance our understanding of cancer initiation and progression at the level of cellular pathways. We have seen in Section 4.6.1 that non-linear classifiers have individual explanations and therefore require a way to aggregate all individual explanations for data points into a model-centric explanation of the important *connectome*, i.e. the parts of the **PPI** interaction graph most relevant for EMOGI.
Section 5.8 (and Figure 5.9 in particular) has introduced that modules in the **PPI** graph can be computed through the aggregation of individual explanations across all genes. To that end, interaction contribution matrices for all $13,627$ genes from the **CPDB PPI** network E^I were subjected to a pointwise summation and the resulting matrix (E^I_{total}) was interpreted as a weighted and directed graph where edge weights correspond to the importance of interactions. This graph is next sparsified and **Strongly-Connected Components** (**SCCs**) are computed within the sparse graph. Those **SCCs** correspond to the most important parts of the **PPI** network and biologically represent cancer gene networks and highly relevant pathways.

7.3.1 *Deriving Edge Weights From Explanations*

The **LRP** rule produces — in addition to the multi-omics contributions — scores for edges in the **PPI** network from the point of view of the gene that the explanation is calculated for (Section 5.8 explains how **LRP** was used for EMOGI). Depending on the local support of the graph convolutional network (the degree of the Chebychev polynomials used, as explained in Section 4.5.3) and the number of graph convolutional layers, those include direct neighbors and indirect connections as well.
From the above-mentioned aggregation — explained in more detail in Section 5.8 — results a directed and weighted graph. The edge weights represent a general importance of **PPIs**, which is reflected by a mild correlation between edge betweenness (the

fraction of shortest paths that pass through the edge) and the importance of the edge in the directed contribution graph (Pearson's correlation coefficient 0.08, depicted in Figure A.14). From that graph, standard community detection algorithms can be used to find modules of nodes with high **PPI** weights.

One such standard algorithm first truncates the graph by removing all edges below a certain threshold and then computes **SCCs** within the truncated network. **SCCs** are defined for directed graphs only and denote sub-graphs in which there exists a path from every node to every other node. **SCCs** have been used previously for the detection of highly mutated cancer modules where a directed network was computed using random walks [41] and can be computed efficiently, even for large graphs [260]. Selecting only a subset of edges is required because biological networks are most often scale-free and hence, redundant paths from all nodes to all others exist [125, 218]. Similarly to the number of clusters for the bi-clustering procedure, the choice of a cutoff for the directed network is not trivial. One way that is used in practice [41] is to decide how many modules at what sizes are reasonable to assume and then select a threshold that produces such numbers of modules. To that end, the threshold that resulting in the highest amount of **SCCs** with size $\geqslant 5$ was selected (depicted in Figure 7.9 along with how the threshold affects the network). Tarjan's algorithm [260] was used to find **SCCs** in the network.

The resulting modules are those parts of the **CPDB PPI** network that EMOGI is most focused on and correspond to a connectome of important pathways, protein complexes or other biological structures that are encoded in the topology of the **PPI** graph.

7.3.2 *Modules of Cancer-Related Protein-Protein Interactions*

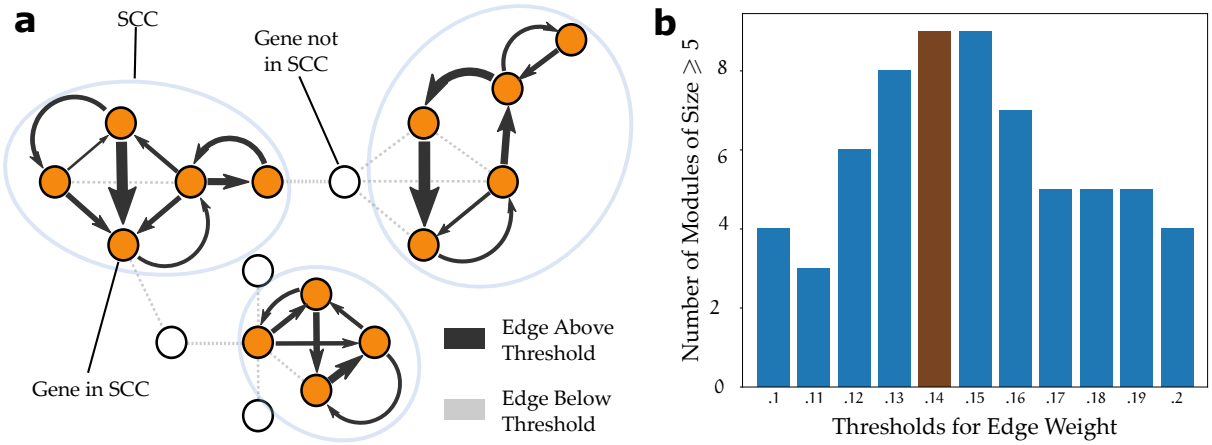


Figure 7.9: **Thresholding removes edges and reveals components of high contributions. a** Removing all edges below a certain threshold makes the **PPI** network more sparse and only **PPIs** with high contributions stay. In such a sparse network, **Strongly-Connected Components** (**SCCs**) can be detected. However, finding a threshold that creates a sparse network with a sufficient number of components is not straightforward. **b** The threshold can be determined by trying out which one yields the highest number of components. Here, the goal was to find meaningful **SCCs** with size $\geqslant 5$. The threshold of 0.14 produces 9 **SCCs** of sufficient size.

With the selected threshold of 0.14 (depicted in Figure 7.9b), 45 **SCCs** with a total number of 323 genes were found. The largest **SCC** contains 149 genes and the smallest one only 3. The average **SCC** size was 3.1, but further analysis was only done for the 8 **SCCs** of size $>= 5$.
Those first 8 components included genes involved in well-known cancer pathways, or complexes that have more recently been observed to be important in cancer, as well as new sub-networks with potential new roles in cancers.

The largest **SCC** corresponds to a big **SCC** of 149 genes and represents the core "interactome" used by the EMOGI model to perform the cancer gene classification task (Figure 7.10). This component is enriched in genes predicted as cancer-related by EMOGI, as well as **KCGs** in cancer pathways, such as focal adhesion, TGF-beta signaling pathway, ECM-receptor interaction, Wnt signaling pathway and ErbB signaling pathway, among others (Table S9, KEGG pathway enrichment analysis). Interestingly, the **SCC** is highly enriched for extracellular matrix genes (according to Gene Ontology enrichment, p-value $= 2.4e^{-10}$, Table S8) which are known to be a major structural component of the tumor microenvironment [261].

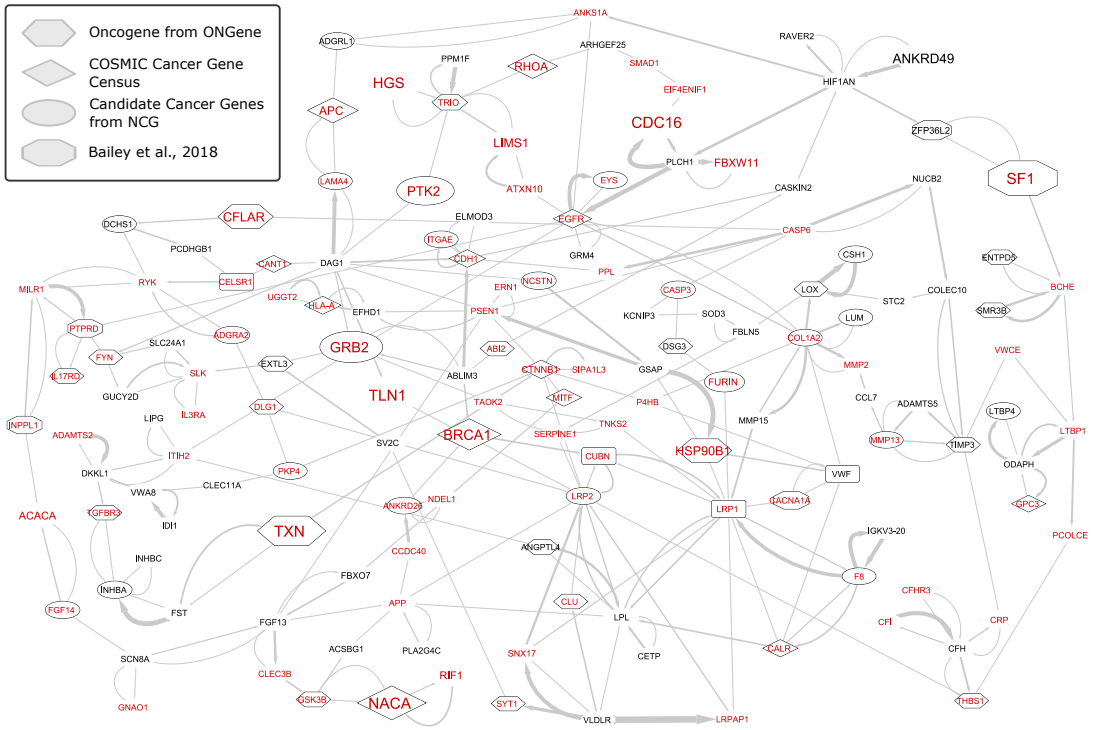


Figure 7.10: **EMOGI allows extraction of PPI network components corresponding to known cancer sub-networks, as well as putative novel cancer modules.** The largest **SCC** of important edges in the **CPDB PPI** network. Red gene names indicate that the gene was predicted as cancer gene by EMOGI, the shape of the nodes indicates presence in a database of cancer genes and the size scales with the number of tumor cell lines in which the gene was essential in the Achilles cancer dependency map (see Section 7.1.3 for details). The width of the edges scales with the importance of the edge for the EMOGI model.

Among the well-known cancer genes that the component revolves around are *BRCA1*, *GRB2* and *CDH1*, mainly associated with breast cancer, the tumor suppressor *COL1A2*, whose altered expression patterns have been linked to the development of colorectal cancer [262] and the ErbB family member *EGFR*, driver of tumorigenesis in mainly lung, breast and brain cancer [263]. *EGFR* interacts with *PTPRD*, a receptor known to regulate oncogenic transformation and cell growth [264]. *PTPRD* in turn interacts with several other genes linked to cancer, such as the Interleukin 17 receptor *IL17RD* or *FYN* — a **NPCG**.

Cell adhesion molecules take part in intercellular and extracellular matrix interactions of cancer, playing a pivotal role in cancer development and metastasis [265]. The largest **SCC** contains several proteins of the endoplasmic-reticulum proteins involved in cell-cell adhesions, such as β-catenin (*CTNNB1*), the blood coagulation factor *F8*, calreticulin (*CARL*) and the heat shock protein *B1*. Other important genes of this component are those forming star-like structures, for example the endocytic cell signaling receptors *LRP1* and *LRP2*, which have been shown to be critically involved in many processes driving tumorigenesis and progression [266], the inflammatory caspases *CASP3* and *CASP5*, which have been shown to regulate apoptotic response [267] and the *TXN* transcription factor, which links a sub-module centered around

the *TGFBR3* tumor suppressor gene to *BRACA1* and its interacting partners.

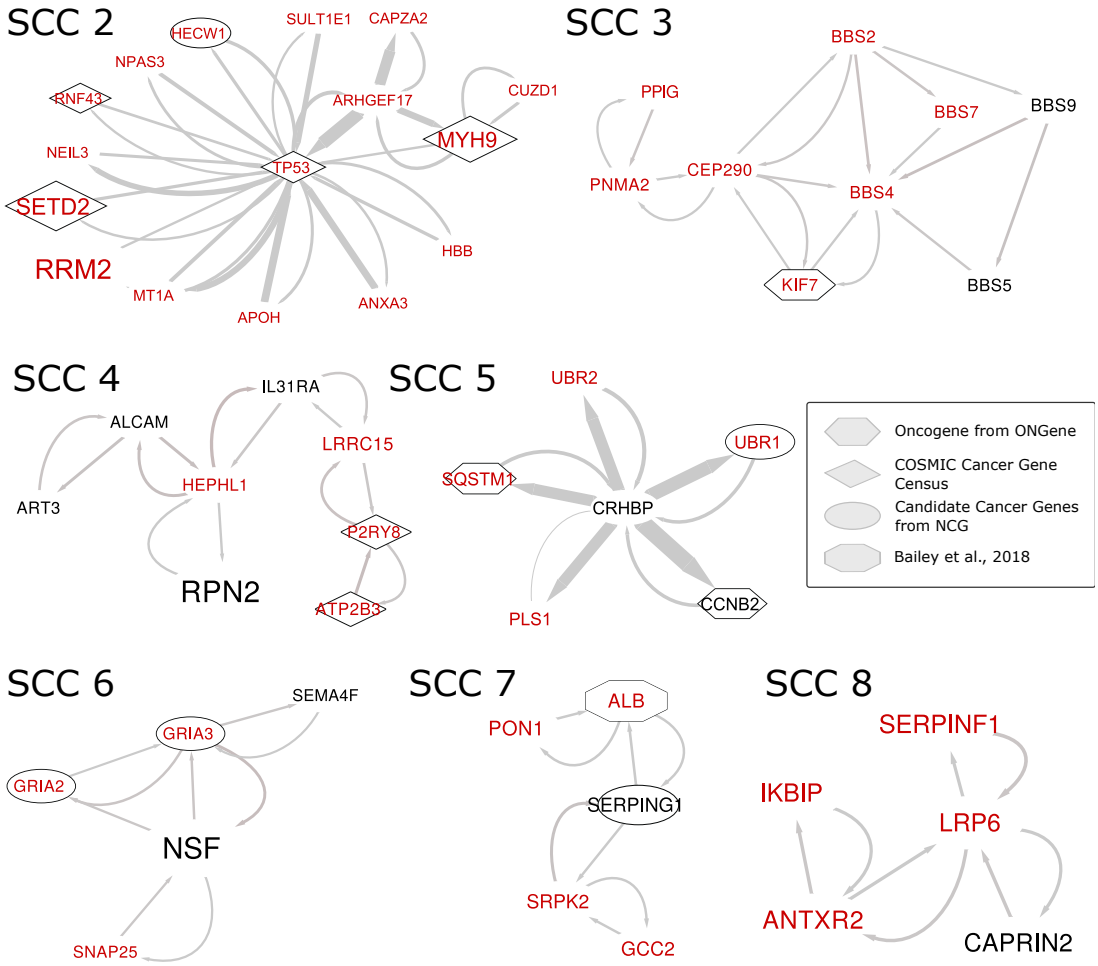


Figure 7.11: **Additional SCCs with five or more genes. SCCs** in a network where edges are directed and weighted, based on the importance of the edge for the classification. The approach identifies 7 additional **SCCs** which are depicted here. The size of the nodes is proportional to the number of tumor cell lines where the gene was found to be essential, according to the Achilles data. Gene names marked in red correspond to genes which were predicted to be cancer genes by EMOGI, and different node shapes indicate whether and in which database the gene was already annotated as cancer gene.

The second-largest **SCC**, depicted in Figure 7.11 along with all other components with five or more genes, contained some well-known cancer regulators, such as the tumor suppressor *TP53*, which forms a star-like structure at the center of the component, its regulator *MYH9*, known to function either as tumor suppressor [268] or oncogene in different cancers [269] and the histone-lysin 36 methyltransferase *SETD2*, a marker of active chromatin and trancriptional elongation recently identified as potential tumor suppressor in solid cancers [270]. Accordingly, P53-signalling is enriched in that **SCC** (KEGG pathway enrichment analysis, p-value $= 0.0035$).
The remaining identified components correspond to complexes that have more re-

cently been observed to be important in cancer, as well as new sub-networks with potential new roles in cancers. For example, the BBsome complex was found, a cargo adapter for many signaling proteins with important roles in cilia homeostasis [271], whose association with cancer is not yet known from the scientific literature (Figure 7.11, SCC 3).
Furthermore, the fifth-largest **SCC** (Figure 7.11) is centered around the **TSG** *CRHBP*, identified as **TSG** due to its role in regulating apoptosis and inflammation [272]. The component appears to be linked to the ubiquitination processes because of the prominent positions of the **NPCGs** *UBR1* and *UBR2*. Ubiquitination is important for cellular homeostasis and alterations in this process lead to various types of cancer [273]. In addition, the fifth-largest **SCC** is linked to the pro-survival NF-κB pathway through the *SQSTM1* gene [274].
A last component of interest, containing only **NPCGs**, is the 8^{th}-largest **SCC** with *LRP6*, a receptor protein in the Wnt/β-catenin signaling cascade that was reported to regulate cell differentiation, migration and proliferation [275]. It interacts with the Proliferation-Associated Protein *CAPRIN2* that also regulates Wnt signaling [276] and *SERPINF1*, a protein that inhibits angiogenesis, the process of growing blood vessels which is highly linked to cancer [277] (Figure 7.11).

7.3.3 *Summary*

Interrogating the direct network derived from the LRP analysis allows us to extract modules corresponding to those parts of the **PPI** network where EMOGI focuses the most, and allows the identification of well-known cancer modules, as well as new complexes with putative undiscovered associations to cancer.
However, as pointed out by another study [151], the averages of gene-wise explanations should not be over-interpreted, as the average may simply point to more central parts of the **PPI** network while individual genes have highly specific local topologies important for their classification as cancer genes. If a module of interactions is highly important for only a few cancer genes, this might not be adequately represented in our module detection because we average over all the genes in the **PPI** network, decreasing the module's impact on the overall analysis. Nonetheless, the identification of such modules complements the analysis of novel findings by EMOGI.

7.4 DISCUSSION

EMOGI is a powerful classification method that integrates heterogeneous data across cancer types. When we examined new findings from the algorithm, we saw that EMOGI found new putative cancer genes, termed **NPCGs**. Examining the **NPCGs** in more detail, we found them interacting with **KCGs**. This is strong evidence that **NPCGs** are part of known cancer protein complexes, pathways or even more general cellular processes. From a computational perspective, the nodes (genes) in the **PPI** network are said to exhibit high *homophily* because neighboring nodes have a higher chance to share the class label. **GCNs** were previously shown to perform well in such applications with high homophily but this finding also explains an observation made

in Section 6.2.5. The high performance of DeepWalk, only trained on the **PPI** network and unable to incorporate knowledge from the *omics* levels exhibited a high performance and outperformed many of the other methods on the **CPDB PPI** network. The high homophily in the network, however, provides an explanation for the observation. DeepWalk conducts short random walks (of length 10) across the network and projects nodes that are often seen together during the walks close to one another. Hence, DeepWalk will fully exploit the homophily and project cancer genes in close proximity. The **SVM** used to classify based on the node embedding is then capable of identifying cancer genes close to each other.

Very interestingly, the **NPCGs** are also more likely to be essential genes in loss-of-function CRISPRi screens and their knock-down leads to reduced growth of cancer cell lines. This observation is surprising because EMOGI was not trained on such data. While it was shown in previous studies that central nodes in **PPI** networks are also more likely to be lethal when removed [93, 214, 278], neither the **PPI** network nor the multi-omics features provide the **GCN** algorithm with a direct measure of essentiality. Because such loss-of-function screens do not exist for non-cancer cell lines, it is hard to argue that all of the essential **NPCGs** are valid drug targets since their knockdown could be lethal for normal cells as well. However, the analysis presented here showed that the **NPCGs** are not solely housekeeping genes, making at least a part of them interesting targets for further studies.

Next, we attempted to find groups of cancer genes driven by similar evidence, thereby fully exploiting the heterogeneous data integration of EMOGI. Following the hypothesis that a gene does not have to be mutated itself in order for it to be associated with cancer, several different classes of cancer genes were found. Some were interactome-driven and required a set of interaction partners to be classified as cancer genes and consequently, most of the **NPCGs** were found in those clusters. Previous studies have identified such "helper genes" in the past [279] and their presence was discussed previously [37].

More traditional groups of cancer genes represent clusters driven by **SNVs** that contain many known cancer genes and less impact from the interactome. Genes from such clusters can probably be found with other computational methods, such as MutSigCV or 20/20+ that analyze mutation frequencies in genes.

CNA-driven clusters were enriched for tumor initiation genes and generally contained more **KCGs** compared to other clusters. Among the copy-number driven genes were *MYC* or *NRAS* that are known to be amplified in multiple cancer types.

A third group of cancer genes is comprised of differentially expressed genes whose altered gene expression might be the result of non-coding mutations or epigenetic changes. Such genes could be interesting to study further because they might represent regulators that the cancer cell is relying heavily upon to gain or maintain its growth advantage in the tissue microenvironment [37]. Accordingly, genes for which gene expression was most important exhibited high essentiality in **CRISPRi** loss-of-function screens.

Lastly, a cluster of genes driven by aberrant DNA methylation was discovered. Examining those genes more closely, we identified **TFs** and immune-related genes in this group of genes.

In addition, the classes of cancer genes further divide into more tissue-specific sub-

clusters, highlighting that the pan-cancer approach is helpful when attempting to find general regulators of cancer diseases. The bi-clustering analysis revealed distinct classes of cancer genes but relied on the multi-omics features for its grouping. To also analyze contributions of individual **PPIs** in more detail, we examined modules of highly important interactions. Here, we found a large module that largely captures the Wnt signaling pathway and contains many well-known cancer genes. This component represents a connectome, a highly relevant module comprised of cancer genes and important regulators of cell proliferation, apoptosis and other pathways relevant for cancer (Table A.5 and Table A.4). The finding highlights that analyzing the contributions of individual **PPIs** in more detail can yield further insights and provide additional putative novel cancer genes.

8

DISCUSSION & CONCLUSION

8.1 DISCUSSION

It is widely accepted that cancer diseases arise through the accumulation of somatic mutations which confer growth advantages enabling cells to grow outside their otherwise tightly regulated microenvironment. The exact molecular changes underlying such transformations, however, are often still unknown. This is partly due to the sheer complexity of cells where redundancies in cellular pathways, as well as a plethora of layers of regulation are responsible for the concentration of particular proteins within the cell. Hence, each tumor is unique with its own resistance to specific drugs and with different growth characteristics.
To nonetheless be able to effectively treat cancer diseases, the long-term goal of personalized medicine attempts to first sequence (or otherwise profile) tumor cells and then guide treatment with specialized drugs targeting the unique characteristics of the profiled tumor cells. However, personalized medicine requires a deep understanding of key regulators which might be mis-regulated in tumors as well as a battery of drugs able to target a wide variety of such regulators. While they do not have to be proteins and drugs targeting **RNAs** have been successfully developed in the past, the majority of personalized cancer drugs attempt to restore or repair protein levels and continue to do so (see Section 2.2.1 for an introduction to precision medicine). The identification of genes that are often mis-regulated in cancer diseases is therefore a crucial point in precision medicine.

Their identification is complicated not only by an exact definition of what constitutes a cancer gene but also by the vast amount of sequencing data generated in high-throughput screens and large consortia like **TCGA** or the **ICGC**. Machine learning methods can help to find abnormalities in large data sets across cohorts but have mainly been used to find highly mutated genes or pathways [29, 41, 42]. However, it is by now widely accepted that while mutations are causing cancer diseases, their effects are often indirect [19, 25, 37] and consequently, the identification of cancer driver genes either requires a complete understanding of gene regulation or the integration of complementary experimental readouts, such as DNA methylation or gene expression profiles.

In this thesis, we saw how EMOGI — a new approach for identifying pan-cancer genes — is able to integrate different data modalities into an explainable machine learning model. EMOGI is based on **Graph Convolutional Networks** and on the integration of connectivity between genes (and their products) in a **PPI** network with features, such as mutation rates, copy number changes, gene expression and DNA methylation changes into a single predictive model.
In two main aspects EMOGI represents a methodological advancement with respect

to previous methods. First, it is able to combine multivariate *omics* data with gene-gene proximity in biological networks, thereby extending previous approaches based on either detection of frequently mutated genes or disease modules from combinations of somatic aberrations. Second, for a machine learning method to be considered trustworthy in clinical applications, one needs to ensure that its decisions are supported by meaningful patterns in the input data. EMOGI is interpretable through the application of the **LRP** propagation rule, allowing to explain individual predictions and to highlight which input features, i.e. *omics* levels or network interactions, are the most important for classification. Interpretability of machine learning models becomes increasingly important when the goal is to gain mechanistic insights in a biological process or in clinical applications, where additional support is needed to build trust into a predictive model.

The lack of a precise definition of what characterizes a gene that associates with cancer diseases leads ultimately to a lack of *bona fide* cancer gene sets, complicating an unbiased evaluation of prediction methods. This is underlined by the low agreement between methods for cancer gene prediction and cancer gene sets ([31], Figure A.6 and Figure A.7).
In the absence of a gold standard set of cancer genes, the performance of EMOGI was evaluated on several cancer gene sets and compared to other methods for cancer gene prediction in Chapter 6. While no method outperformed all others in all settings, EMOGI performed best on average, achieving between 3% and 37% higher **AUPRC** values compared to the other tools. EMOGI's performance was also very stable across different cancer gene data sets, while the performance of other methods, mainly Random Forests and 20/20+, were highly dependent on the data set. This most likely reflects different biases in data set collection and highlights once more that different assumptions about cancer driver genes lead to different results. The data set from Bailey et al. [198], for instance, includes cancer genes mainly predicted from mutation rates and consequently, MutSigCV and 20/20+ performed very well on this data set. The deep learning method DeepWalk was among the methods that exhibited, on average, the highest performance after EMOGI, despite the fact that it uses only network topology and not the *omics* features for cancer gene prediction. This observation suggests that recently compiled **PPI** networks encode the main properties of known cancer genes in their topology, but it also points towards a study bias where well-known cancer drivers, such as *KRAS* and *TP53* have been more intensively studied and therefore more of their interaction partners in PPI networks are known [217]. Furthermore, as already discussed briefly in Section 7.4, the high performance of DeepWalk also implies high homophily in **PPI** networks such as **CPDB** or STRING-db, where two interacting genes have a much higher chance to share the same label (e.g. cancer gene or not). Consequently, network propagation and similar methods can only hope to identify putative cancer genes from pathways and protein complexes of which parts have previously been annotated if they start exploration from a set of known cancer genes.

Perturbation experiments from Section 6.5 showed that the integration of both network and *omics* features was crucial to achieve a high classification performance and

that the use of multi-omics data boosted EMOGI's performance compared to the case when only a subset of *omics* levels was used for training. The fact that the mutation rate accounted for most of the method's performance probably reflects the over-representation of highly mutated genes versus genes harboring other types of alterations in the training and test sets but is also likely to be biologically meaningful as cancer malignancies are ultimately a set of diseases of the genome. The multi-omics setting is further likely to increase the certainty in individual predictions, as can be seen from the overlap between predictions for *omics* types. EMOGI recovered 89% of the known cancer genes and contains well-known cancer drivers among its top predictions, such as *PTEN*, *PIK3CA*, *AKT1*, *APC*, *KRAS*, *TP53* and others. Coupled to the **LRP** feature importance analysis, EMOGI was able to differentiate cancer gene predictions based on their main molecular contributions, i.e. mutation rate-driven predictions versus other molecular alterations, such as copy number changes, DNA methylation and gene expression. The **LRP** framework also identifies important interaction partners of genes and the manual inspection of those revealed important cancer pathways and protein complexes in Section 6.4, such as the RB1-E2F1-HDAC1 complex, the PIK3 signaling pathway and the SWI/SNF protein complex, corresponding to evidence for their molecular mechanisms in the scientific literature [41, 50, 239, 280].

A comprehensive list of 165 **Newly Predicted Cancer Genes** (**NPCGs**) was derived in Section 7.1 and found to be sensible by various measures. All of the **NPCGs** interact with at least one known cancer driver (Section 7.1.2) and there is a clear enrichment of essential genes within the **NPCGs**, with 70% of the novel predictions being essential for the survival of at least five tumor cell lines (Section 7.1.3). While there is an enrichment of **NPCGs** with housekeeping genes, a significant fraction of **NPCGs** show tissue-specific expression and functional enrichment analysis finds signaling and other cancer-related pathways enriched. In addition, **NPCGs** are classified as such primarily because of their network features, i.e. interactions with known cancer genes, and only to a lesser extent because of their mutation rate or other *omics* features, according to the **LRP** explanations. One of the most highly ranked **NPCG**, *YWHAZ*, is not listed in the cancer gene databases used throughout the thesis but has been recently associated with hallmarks of cancer due to its overexpression in leukemia and gastric cancers [281]. This example and many others strongly suggest that EMOGI is able to propose new candidate cancer genes for further experimental validation and that novel predictions represent a set of genes that contributes to the formation and/or maintenance of tumors while not being always subjected to genetic modification themselves.

Bi-clustering of genes and individual **LRP** contributions across cancer types identified subgroups of cancer genes characterized by distinct sets of molecular alterations and provided a summary of the different strategies that EMOGI implemented to classify cancer genes. The algorithm distinguishes clusters which consisted predominantly of genes were the network topology had a stronger effect on the classification decision compared to the *omics* features and vice versa. Furthermore, it differentiated groups of genes whose classification was driven mainly by **SNVs** and groups where

CNAs, alteration of DNA methylation or gene expression in subsets of cancers was the strongest determinant for classification.
These results question one of the most widely used definitions of a cancer driver gene proposed by Vogelstein et al. [25], who defines it as a gene that increases cell growth when somatically mutated. Although this holds true for the majority of genes and EMOGI's predictions exhibit mutation frequency as the most important *omics* feature (as seen in Section 6.6), it is now established that the transformation of a cell to a cancer cell can be achieved through many different ways, not only through mutations and copy number changes targeting the gene itself, but also through epigenetic mechanisms, such as promoter DNA methylation [37] or non-coding mutations in regulatory regions which indirectly activate or silence other genes [19, 41, 282]. Mourikis et al. identified hundreds of "helper" genes which, unlike cancer drivers that harbor recurrent alterations, are less frequently or barely mutated but localize in close proximity to known cancer genes in regulatory networks [279]. With the discovery of interactome-driven cancer genes, this analysis further supports such findings.
Another important benefit of the bi-clustering could be the identification of false-positive predictions and artifacts in the data. Such an application of the **LRP** rule was done in [151], where spectral clustering of contributions in image data was used to quickly assess if some images rely on artifacts. In the application presented here, one could speculate if the high number of **CNAs** in kidney renal carcinomas observed for some genes (Figure 7.6) might contain errors or hyper-mutated samples that were not previously annotated. With the difficulty in preprocessing *omics* data and the high noise inherent to the experimental protocols, such a methodology could provide a valuable tool for finding artifacts and false positive predictions.
Finally, the individual **LRP** values for the gene-gene interactions from the **PPI** network permitted to pinpoint those part of the interactome where EMOGI is focusing on, allowing to identify important cancer-related complexes and pathways.

All of these individual results bring us closer to a more fine-grained definition of what a cancer driver gene can look like, and makes us appreciate the vastly different ways in which a gene can influence cancer cell growth. Previous studies have examined some of these ways individually. HotNet(2) [41] finds cancer modules based on interaction data and proximity of cancer mutations, MutSigCV [40] or 20/20+ [31] predict highly mutated genes, and a recent method integrates multiple *omics* to identify modules of cancer genes, but not individual biomarkers [49]. Being able to explain the classifications, however, allowed to dissect different classes of cancer genes, as well as shared and complementary mechanisms for subgroups of genes for the first time.

8.2 OUTLOOK

The EMOGI framework proposed here is quite general, as it can integrate any type of *omics* data, other than those used for this study, as well as different transcriptional and post-transcriptional regulatory networks. Therefore, the method can be used outside of the cancer genomics field and be applied to study other complex diseases, where multi-omics data are available and functional connections between genes are

relevant to the classification of disease genes.
Due to the averaging of features across patients for a given cancer type, the analysis presented in this thesis is blind to distinct sub-populations within a cancer tissue. However, EMOGI can be easily adapted to perform disease classification directly at sample or patient level and then use the **LRP** importance analysis to stratify patients based on the learned classification features, providing an important analysis tool for future applications in precision oncology and beyond. The patient-wise model presented in Section 6.7 represents a first step in that direction and serves as a proof-of-concept that such a patient stratification can be done with high performance. Although the implementation of a patient-wise model was already done, a detailed evaluation of such an application to a specific cancer subtype and the identification of distinct survival groups of patients remains future work.

8.3 CONCLUSION

In this thesis, we saw how the integration of heterogeneous molecular data sets improves the prediction of cancer driver genes. EMOGI is capable of identifying genes that are missed by tools that only use individual data types and is further able to make use of a interpretability framework to dissect the molecular mechanisms leading to individual classifications. With the advent of large data sets in the field, computational approaches that integrate all available data sets are expected to yield new insights into cancer biology and guide the development of personalized treatments in the near future.
We hope that the novel predictions open avenues for cancer research to focus on the backbone of interaction partners in important pathways. This would not only offer ways to target proteins that have not been in the focus of current research but might allow for drugs applicable to multiple cancer types.
We believe that the classification of genes into different subclasses of cancer genes might open new therapeutic avenues. For instance, non-mutated genes that become crucial for cancer cell survival through other means might be easier to target than signaling genes or transcription factors.

A

APPENDIX

A.1 THE DECOMPOSITION OF THE BIAS-VARIANCE TRADEOFF

Let y denote the true labels of a data set X and let $\hat{y}$ denote an estimate of these labels by a **ML** model. Bias and variance can be written as:

$$Bias(\hat{y}) = \mathbb{E}[\hat{y} - y] \tag{A.1}$$

$$Var(\hat{y}) = \mathbb{E}\Big[\big(\hat{y} - \mathbb{E}[\hat{y}]\big)^2\Big] \tag{A.2}$$

Then the mean-squared error (MSE) can be written as:

$$MSE(\hat{y}) = \mathbb{E}\big[(\hat{y} - y)^2\big] = \mathbb{E}[\hat{y}^2] - 2y\mathbb{E}[\hat{y}] + y^2 \tag{A.3}$$

$$= \mathbb{E}[\hat{y}^2] - 2\big(\mathbb{E}[\hat{y}]\big)^2 + \big(\mathbb{E}[\hat{y}]\big)^2 + \big(\mathbb{E}[\hat{y}]\big)^2 - 2y\mathbb{E}[\hat{y}] + y^2 \tag{A.4}$$

$$= \mathbb{E}[\hat{y}^2] - 2\mathbb{E}\Big[\hat{y}\big(\mathbb{E}[\hat{y}]\big)\Big] + \mathbb{E}\big(\mathbb{E}[\hat{y}]\big)^2 + \big(\mathbb{E}[\hat{y}] - y\big)^2 \tag{A.5}$$

$$= \mathbb{E}\Big[\big(\hat{y} - \mathbb{E}[\hat{y}]\big)^2\Big] + \big(\mathbb{E}[\hat{y}] - y\big)^2 \tag{A.6}$$

$$= Var(\hat{y}) + Bias(\hat{y})^2 \tag{A.7}$$

A.2 DATA PREPROCESSING

***TCGA** Studies and Sample Numbers*

		DNA Meth.		Somatic Mut.		Gene Expr.			Incl.?
Study Name	Cancer Type	Tumor	Normal	TCGA	Synapse	Tumor	Normal	GTEx	
BRCA	Breast invasive carcinoma	796	96	1044	Yes	1109	113	218	Yes
GBM	Glioblastoma multiforme	153	2	396	Yes	169	5	1403	No
OV	Ovarian serous cystadenocarcinoma	10	-	433	Yes	379	-	108	No
LUAD	Lung adenocarcinoma	475	32	569	Yes	535	59	374	Yes
UCEC	Uterine Corpus Endometrial Carcinoma	439	46	542	Yes	552	35	90	Yes

KIRC	Kidney renal clear cell carcinoma	324	160	339	Yes	538	72	36	Yes
HNSC	Head and Neck squamous cell carcinoma	530	50	510	Yes	502	44	70	Yes
LGG	Brain Lower Grade Glioma	534	-	513	Yes	529	-	-	No
THCA	Thyroid carcinoma	515	56	496	Yes	510	58	355	Yes
LUSC	Lung squamous cell carcinoma	370	42	497	Yes	502	49	374	Yes
PRAD	Prostate adenocarcinoma	503	50	498	Yes	499	52	119	Yes
SKCM	Skin Cutaneous Melanoma	472	2	470	Yes	470	1	974	No
COAD	Colon adenocarcinoma	315	38	433	Yes	480	41	203	Yes
STAD	Stomach adenocarcinoma	395	2	441	Yes	375	32	204	Yes
BLCA	Bladder Urothelial Carcinoma	419	21	412	Yes	414	19	11	Yes
LIHC	Liver hepatocellular carcinoma	380	50	375	No	374	50	136	Yes
CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma	309	3	305	Yes	306	3	11	Yes
KIRP	Kidney renal papillary cell carcinoma	275	45	288	Yes	288	32	36	Yes
SARC	Sarcoma	265	4	255	No	263	2	621	No
LAML	Acute Myeloid Leukemia	140	-	149	Yes	151	-	456	No
ESCA	Esophageal carcinoma	186	16	184	No	162	11	790	Yes
PAAD	Pancreatic adenocarcinoma	185	10	183	Yes	178	4	197	No
PCPG	Pheochromocytoma and Paraganglioma	181	3	179	No	180	3	-	No

READ	Rectum adenocarcinoma	99	7	158	Yes	167	10	173	Yes
TGCT	Testicular Germ Cell Tumors	150	-	150	No	150	-	203	No
THYM	Thymoma	124	2	123	No	119	58	-	No
KICH	Kidney Chromophobe	66	-	66	No	65	24	36	No
ACC	Adrenocortical carcinoma	80	-	92	No	79	-	159	No
MESO	Mesothelioma	87	-	83	No	86	-	-	No
UVM	Uveal Melanoma	80	-	80	No	80	-	-	No
DLBC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma	48	-	37	No	48	-	-	No
UCS	Uterine Carcinosarcoma	57	-	57	No	56	-	90	No
CHOL	Cholangio Carcinoma	36	9	51	No	36	9	-	No
SUM		8998	746	10408	20	10351	786	7447	16

Table A.1: **Cancer types and data availability.** The table shows the available number of samples for different *omics* levels for tumor and normal tissues. The last column indicates if the study was included in the pan-cancer analysis presented in this thesis. Included were all studies for which gene expression and DNA methylation data was available for normal and tumor tissues in sufficient quantities and further somatic mutation data was available.

MA plots for Gene Expression Data

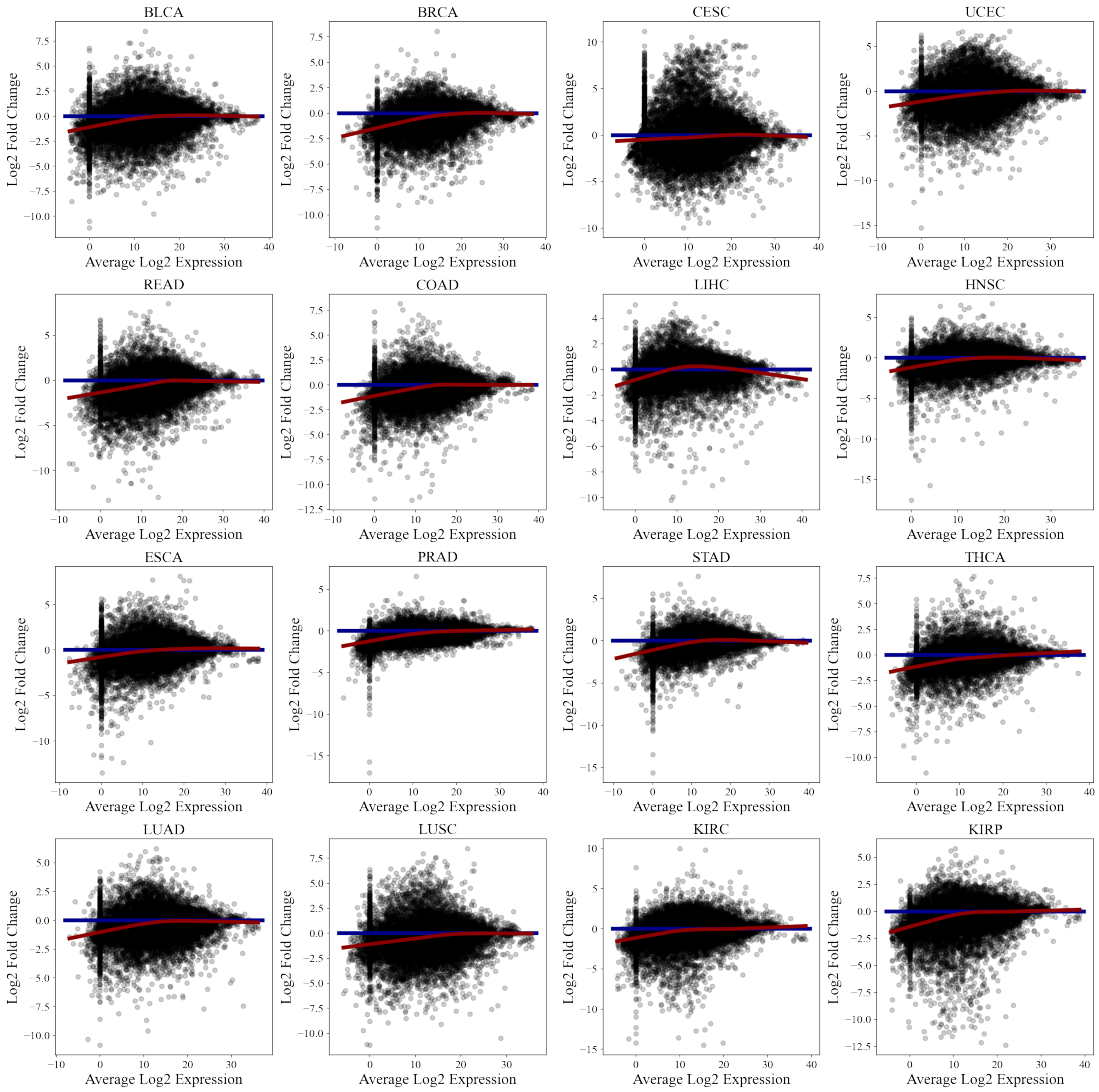


Figure A.1: **MA plots between tumor and normal gene expression values.** For each cancer type, the MA plot depicts the log2 fold change against the average expression for every gene. Point in the plots therefore correspond to genes and differentially expressed genes are found away from the blue line (representing 0). The red line depicts a Loess regression of the data.

UMAP Plots for Samples and Omics Levels

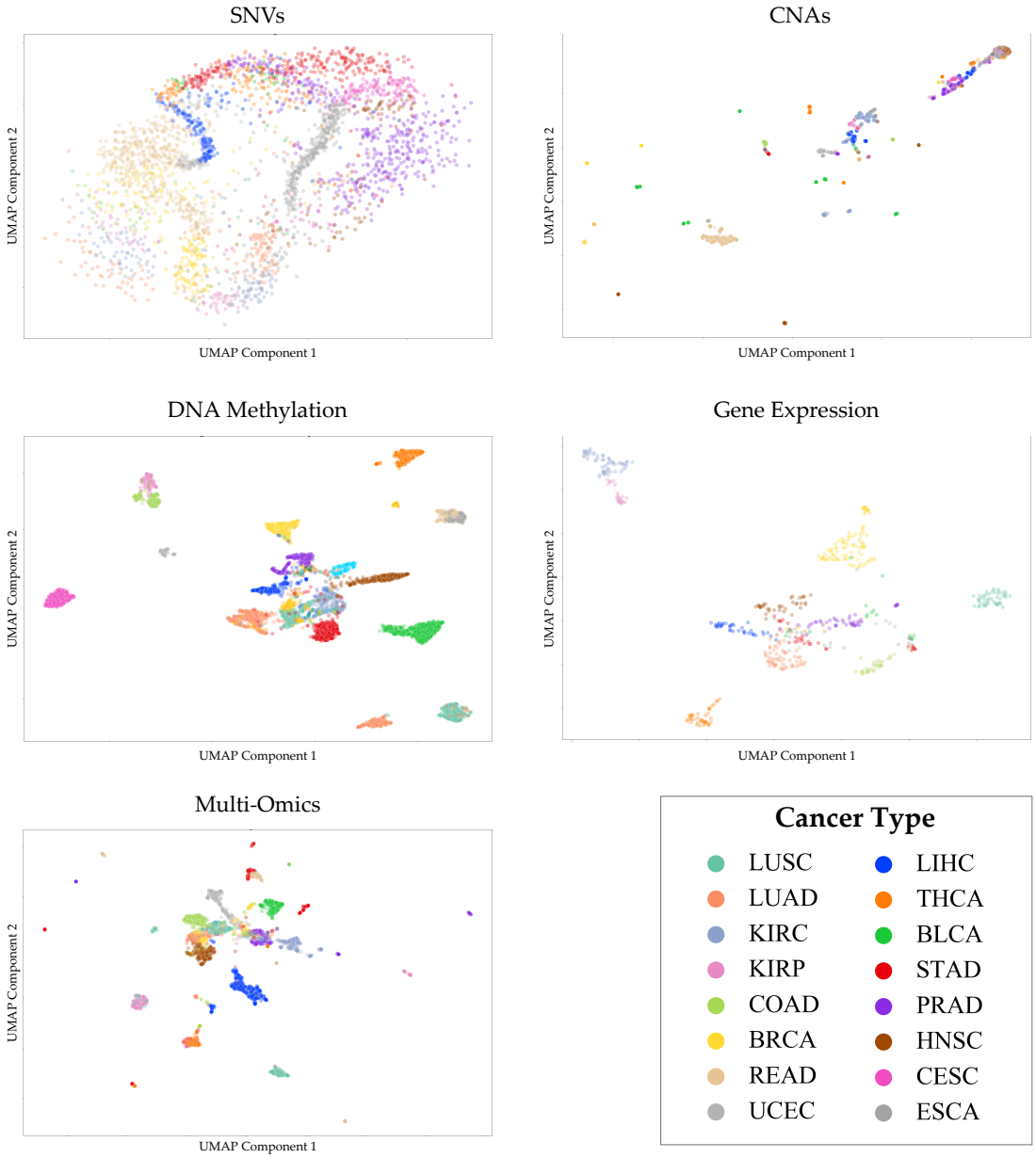


Figure A.2: **UMAP embeddings of the input data types.** For each of the *omics* levels, an UMAP embedding [283] was computed where each point corresponds to a patient/sample from **TCGA** and the color denotes the cancer type that the sample is associated with. UMAP computes a two-dimensional representation of the high-dimensional feature space corresponding to a gene times sample matrix containing the different *omics* features.

The Distribution of Preprocessed & Normalized Feature Vectors

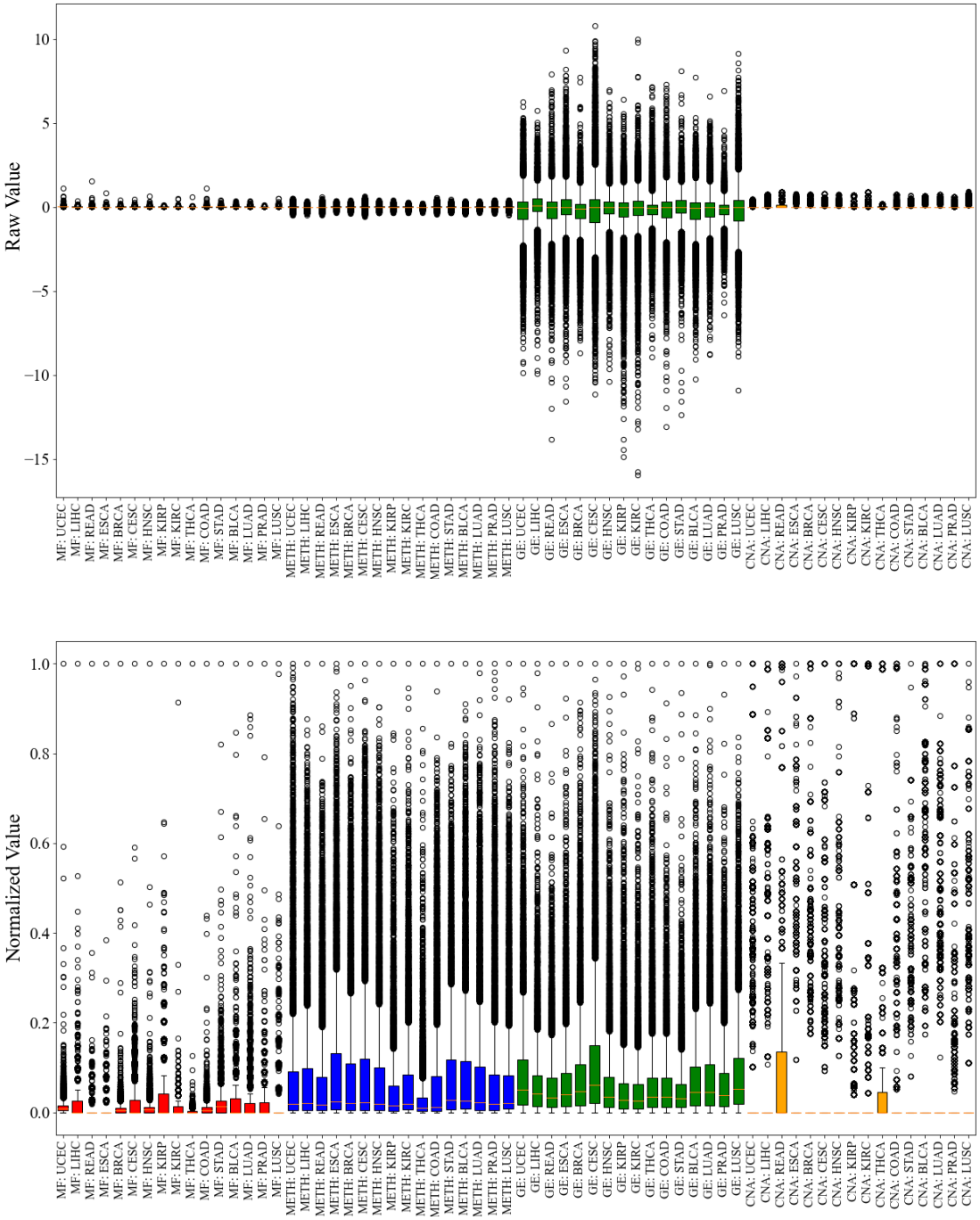


Figure A.3: **Distribution of multi-omics features before and after min-max normalization.** The colors represent the *omics* levels.

A.3 DEGREE BIAS IN THE INPUTS TO EMOGI

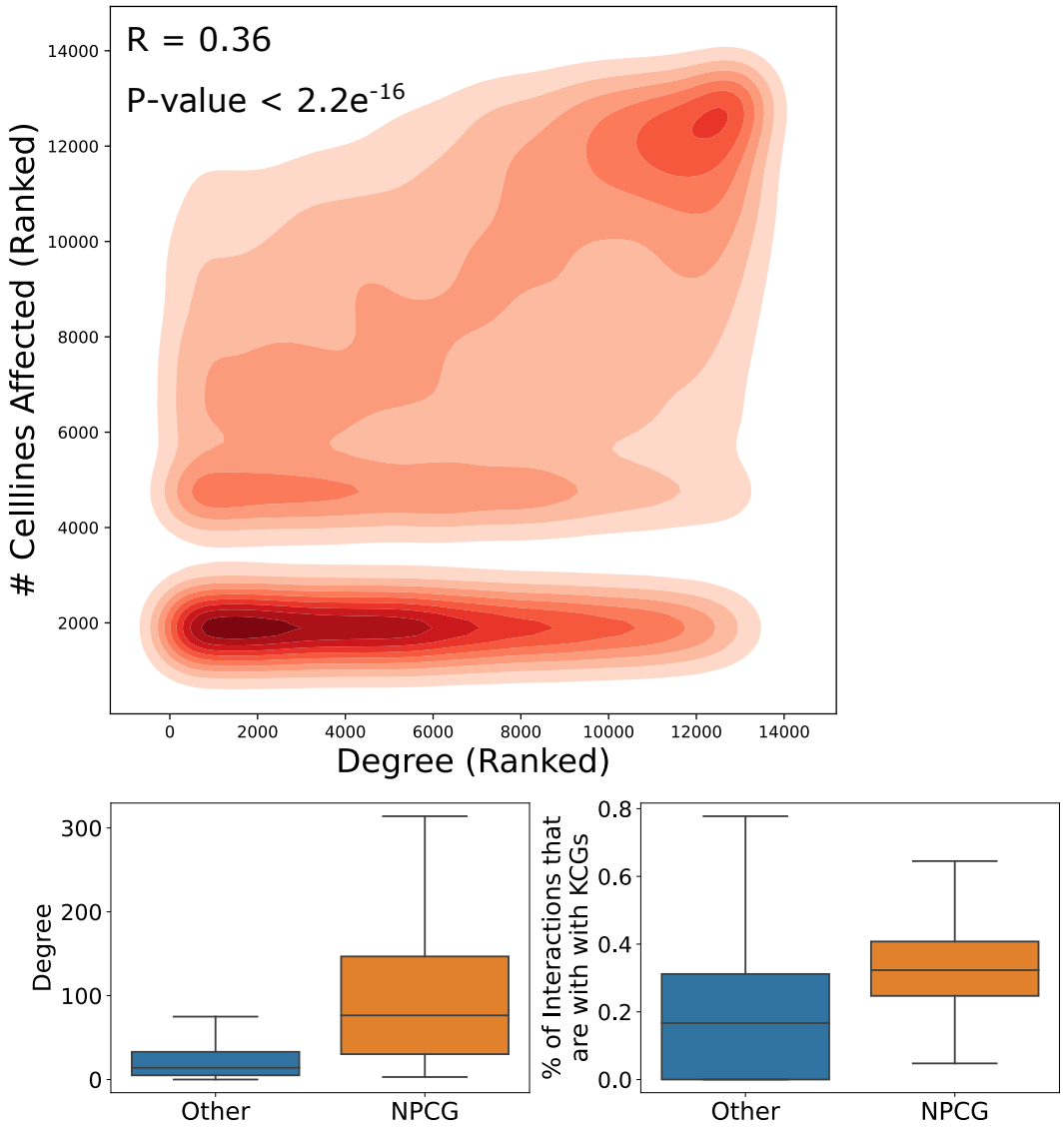


Figure A.4: **Node degree bias in training data and PPIs.** The contour plot depicts the correlation between the essentiality in **CRISPRi** loss-of-function screens with the node degree. Both measures were ranked and spearman correlation was computed ($R = 0.36$). The boxplots show the distribution of node degree for **NPCGs** and other genes, as well as the distribution of interactions with **KCGs** for **NPCGs** and other genes.

A.4 PERFORMANCE EVALUATION

Performance of Additional Network Metrics

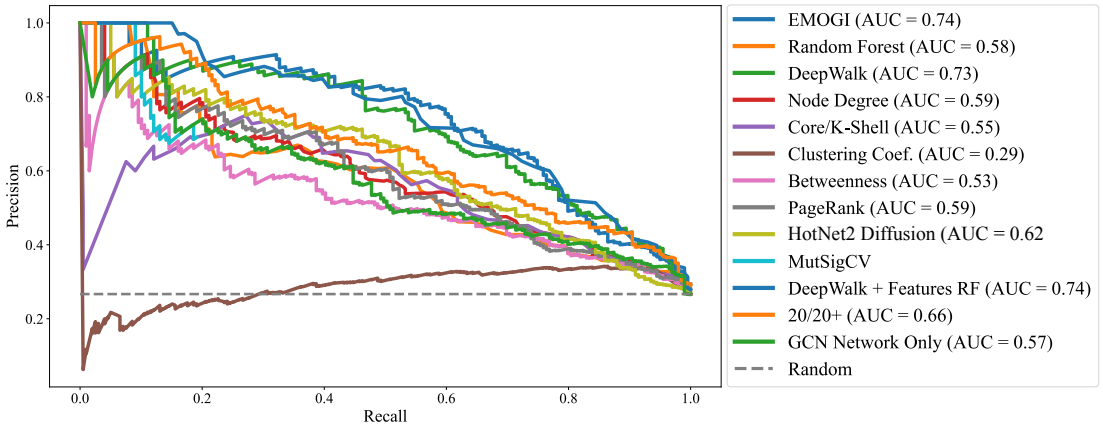


Figure A.5: **PR curve of competing methods and additional network metrics on the CPDB PPI network.** Different additional metrics often used in network analysis are added to the comparison of the methods introduced in Section 6.2.5 and EMOGI.

Overlap between Predictions for Tools

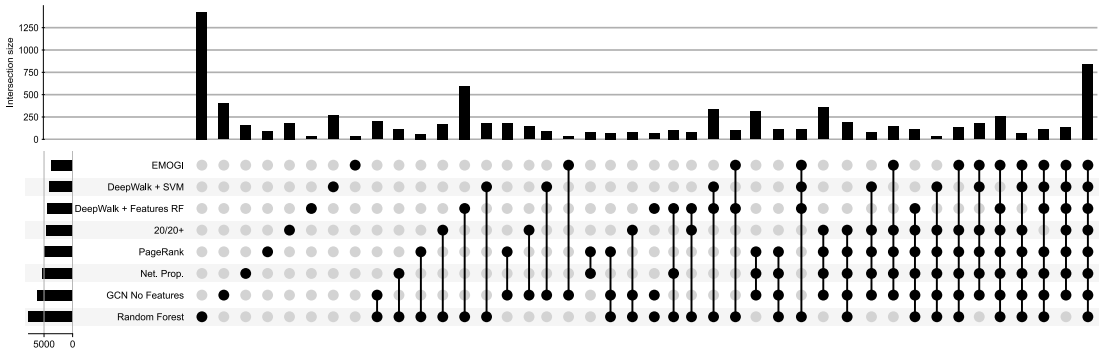


Figure A.6: **Overlap between predictions for various computational methods.** The upsetplot depicts the relevant intersections (for visualization purposes, combinations with no or very low overlap were removed) between the predicted genes for the 7 competing methods and EMOGI. MutSigCV was left out of this comparison because of the few predicted cancer genes from that method. Thresholds on the probability cutoffs were computed for all methods individually, based on the intersection between precision and recall (see Figure 6.5 for details on the computation of optimal thresholds).

A.5 VALIDATION ON INDEPENDENT CANCER GENE SETS

Overlap between Cancer Gene Sets

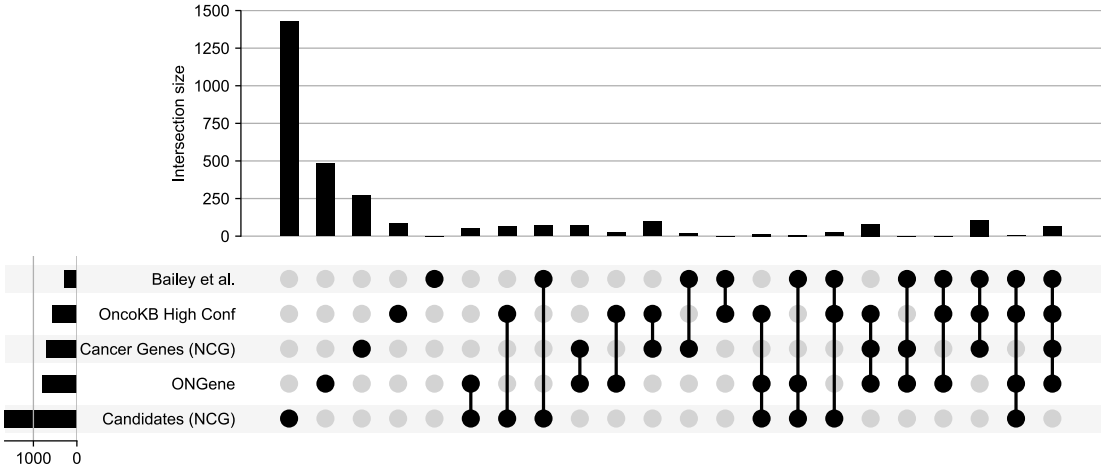


Figure A.7: **Overlap between cancer gene sets used throughout the thesis.**

A.6 THE LRP EXPLANATIONS FOR KNOWN CANCER GENES AS MODEL VALIDATION

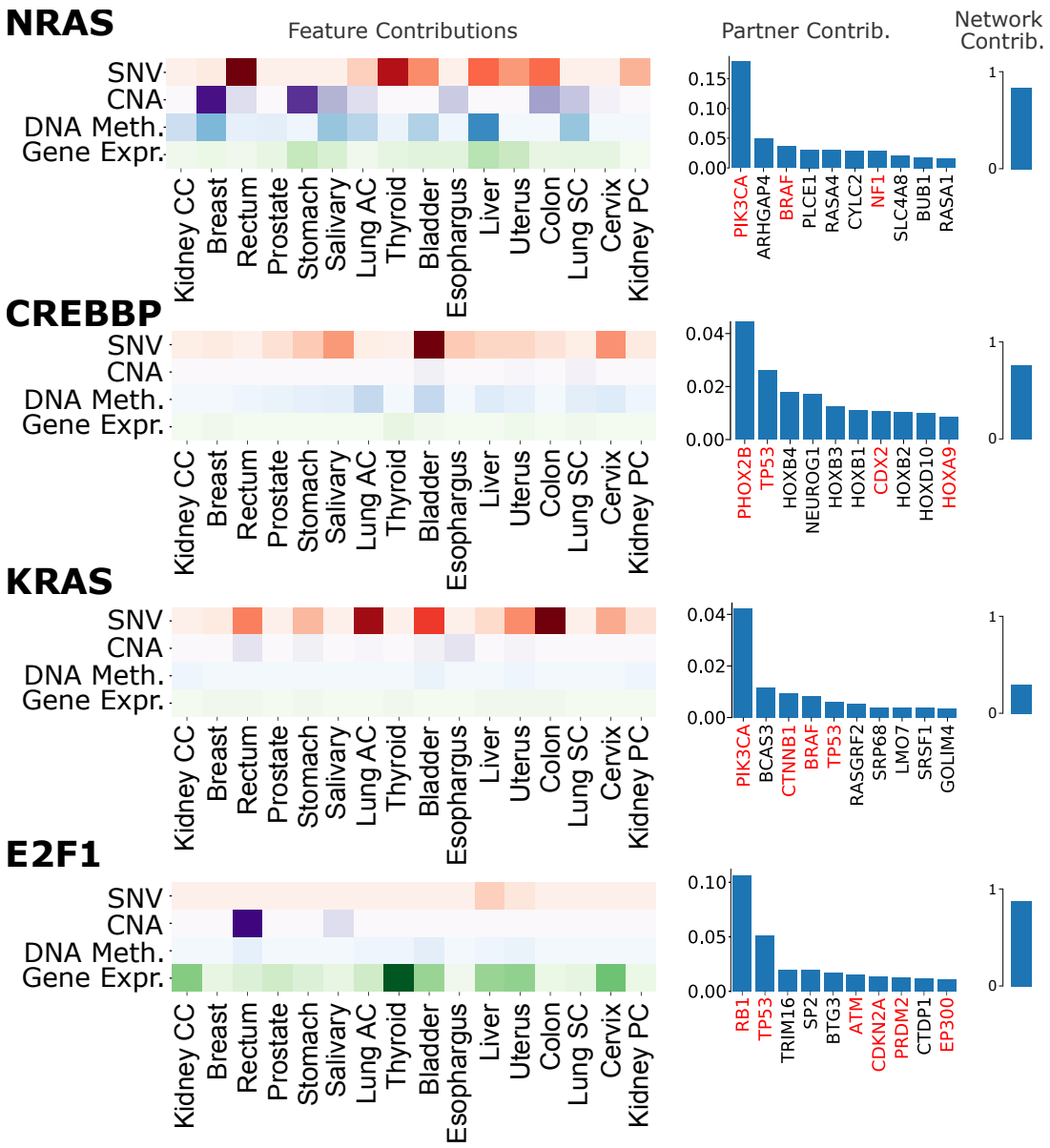


Figure A.8: **Further explanations for KCGs on the CPDB PPI network** Depicted are important *omics* features, as well as the most important **PPI** interaction partners computed through the **LRP** technique for four additional selected genes (see Section 6.4 for details). Heatmaps depict the importance of individual cancer types for different *omics* levels (darker colors indicate higher importance). The bar plots (middle column) depict the 10 most important interaction partners for the gene of interest. The scale of the bars denotes how much of the overall EMOGI score originates from the interaction with that gene. The right column depicts the amount that the interactome contributes to the classification of the gene of interest and ranges from 1 (only the PPIs was important) to 0 (only the features were important).

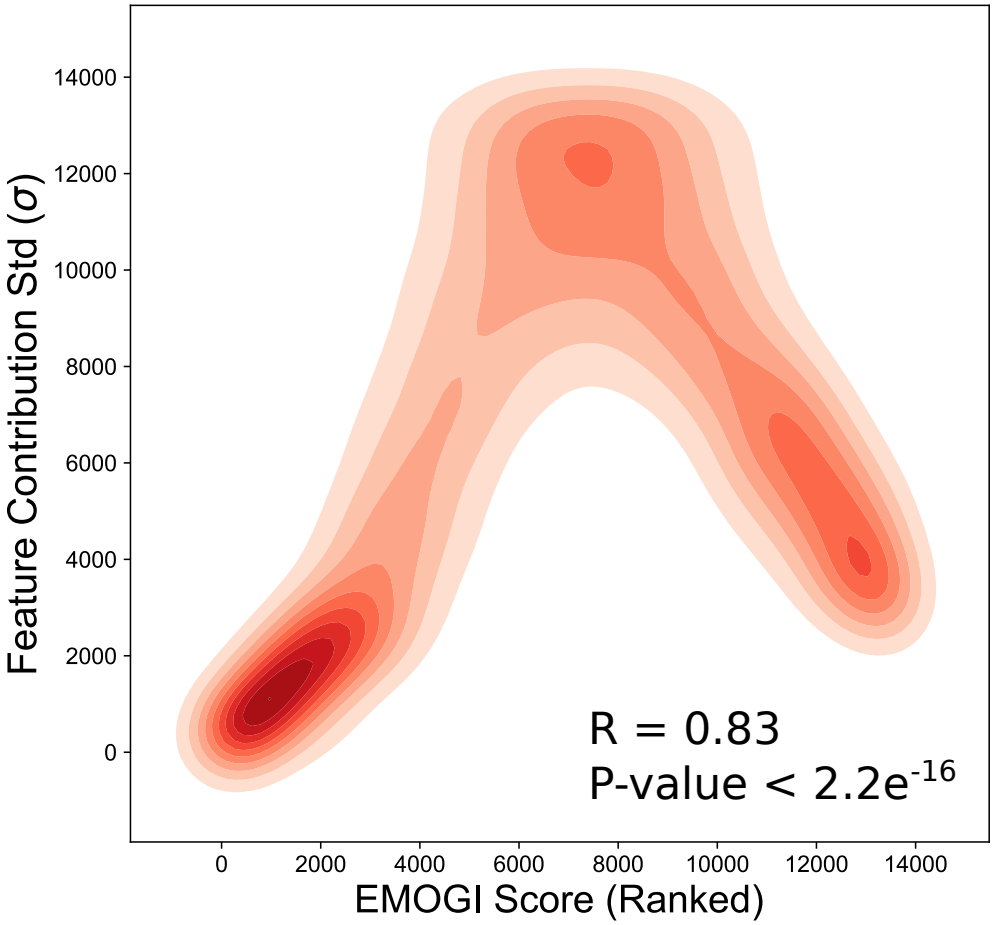


Figure A.9: **Correlation between LRP feature contributions Std and EMOGI score.** Spearman correlation depicted as contour plot for all 13,627 genes of the **CPDB PPI** network. Darker colors correspond to higher density. Both, EMOGI score and the Std of LRP feature values were ranked and higher ranks correspond to higher Std or EMOGI scores.

A.7 PERFORMANCE ON A SUBSET OF *omics* LEVELS

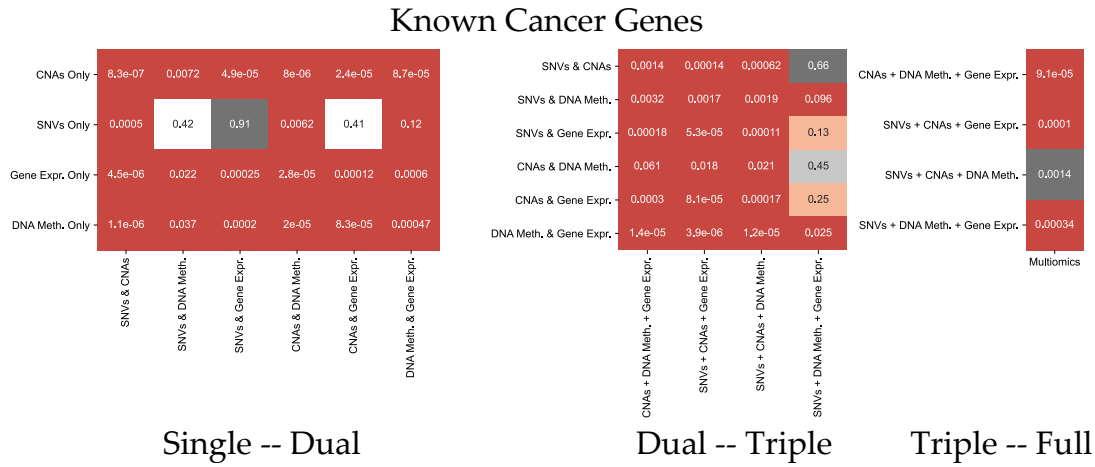


Figure A.10: **T-tests for EMOGI models with increasing *omics* data for KCGs.** The heatmap depicts the results of two-sided t-tests for all pairwise combinations of models when the amount of data for the features is increased. (left) Transition between one data type and two data types. The **SNV**-only EMOGI models achieve partly **AUPRC** values comparable to models using two *omics* levels. (middle) Transition between two *omics* levels and three. Models with increased data perform significantly better, apart from the the model without **CNAs** information. (right) The full multi-omics model performs better than the models only using three *omics* levels.

A.8 FEATURE PERTURBATIONS

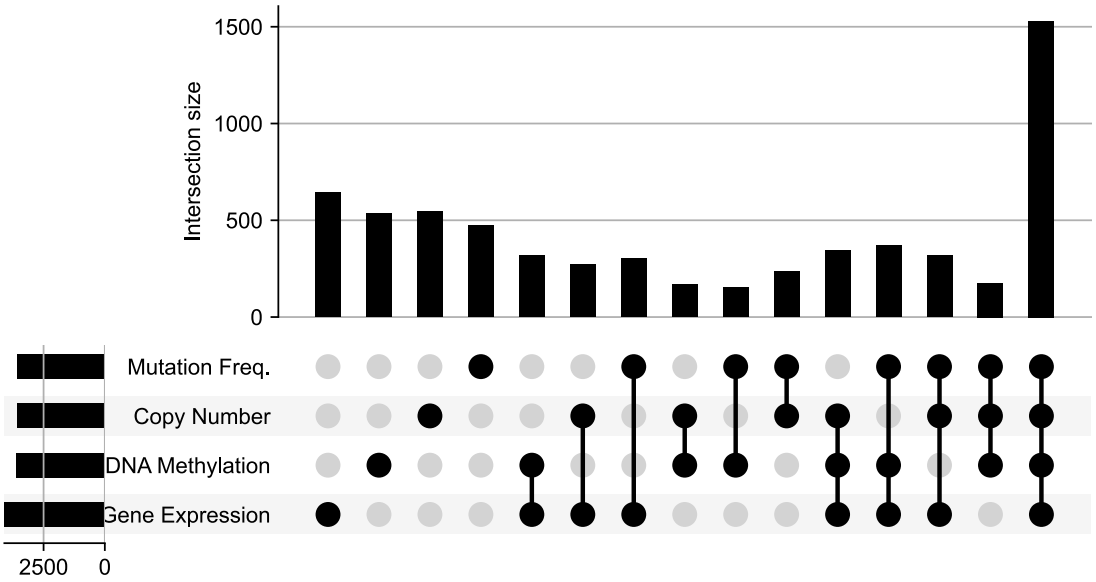


Figure A.11: **Overlap between single *omics* level EMOGI predictions on the CPDB PPI network.** The upsetplot depicts the overlap between EMOGI models that were trained on only **SNVs**, **CNAs**, DNA methylation in promoters or gene expression in comparison to the multi-omics model. Threshold to select predicted genes were calculated based on the intersection between precision and recall for each model separately.

A.9 NEWLY PREDICTED CANCER GENES

Contributions from Individual PPI Networks

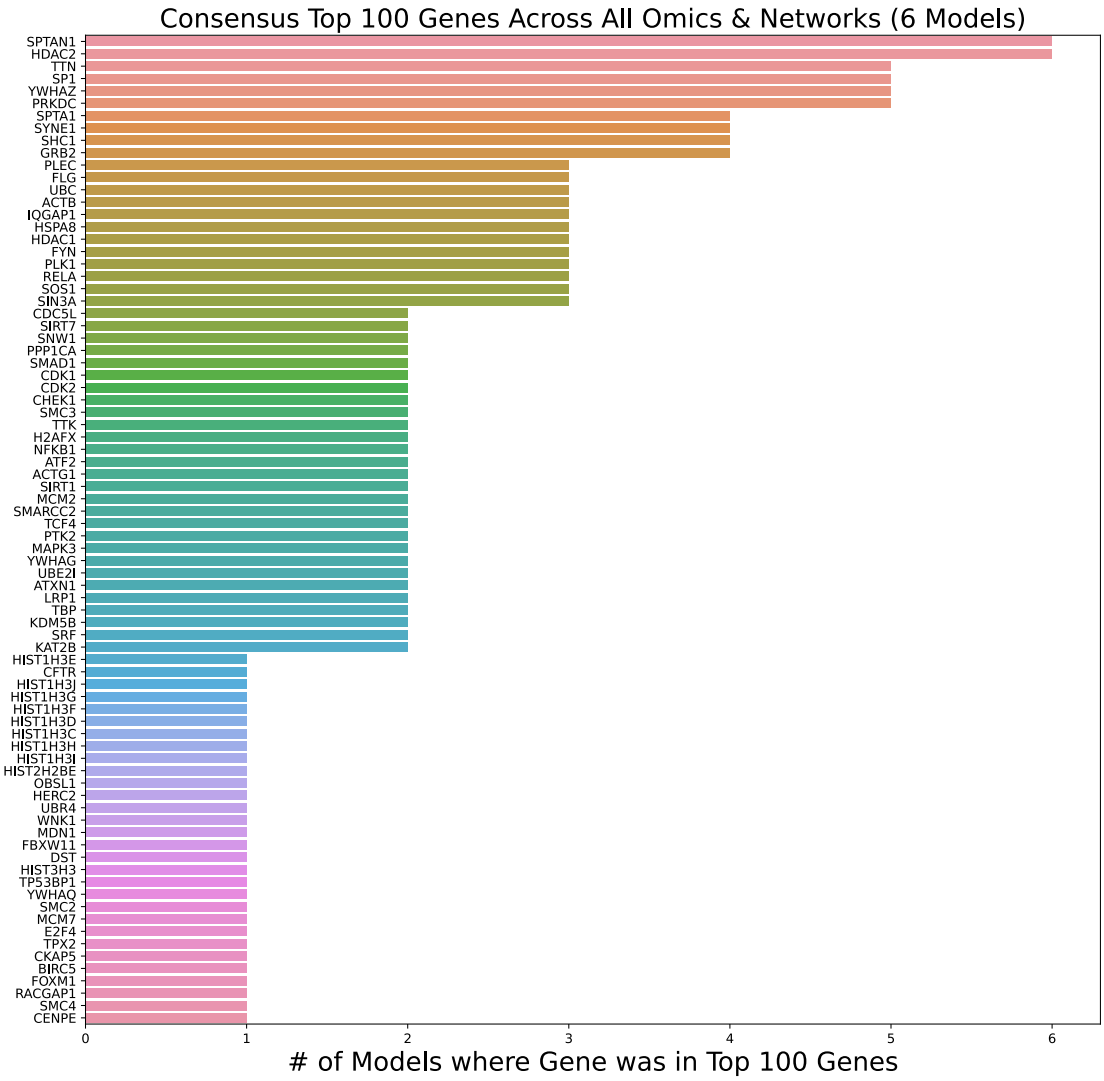


Figure A.12: **Top 80 NPCGs.** For each gene, the bar shows in how many EMOGI models (for 6 different **PPI** networks) the gene was among the top 100 predictions.

*List of all **NPCGs***

Name	Models_top100	NCG_Candidates	OncoKB	Affected_Celllines
SPTAN1	6	True	False	21
HDAC2	6	True	False	13
TTN	5	True	False	8
SP1	5	False	False	130
YWHAZ	5	False	False	121

PRKDC	5	False	True	0
SPTA1	4	True	False	0
SYNE1	4	True	False	0
SHC1	4	False	False	38
GRB2	4	True	False	527
PLEC	3	True	False	42
FLG	3	True	False	0
UBC	3	False	False	612
ACTB	3	True	True	465
IQGAP1	3	False	False	10
HSPA8	3	False	False	187
HDAC1	3	False	True	26
FYN	3	True	True	0
PLK1	3	False	False	625
RELA	3	True	False	188
SOS1	3	True	True	122
SIN3A	3	True	False	578
CDC5L	2	False	False	625
SIRT7	2	False	False	17
SNW1	2	False	False	625
PPP1CA	2	False	False	460
SMAD1	2	False	False	0
CDK1	2	False	False	625
CDK2	2	False	False	553
CHEK1	2	False	True	624
SMC3	2	True	True	625
TTK	2	True	False	616
H2AFX	2	False	False	471
NFKB1	2	False	False	11
ATF2	2	False	False	0
ACTG1	2	True	True	201
SIRT1	2	False	False	5
MCM2	2	False	False	625
SMARCC2	2	False	False	4
TCF4	2	False	False	5
PTK2	2	True	False	429
MAPK3	2	False	True	7
YWHAG	2	False	False	23

UBE2I	2	False	False	620
ATXN1	2	True	False	8
LRP1	2	True	False	12
TBP	2	True	False	352
KDM5B	2	False	False	0
SRF	2	False	False	463
KAT2B	2	False	False	0
HIST1H3E	1	False	True	17
CFTR	1	False	False	2
HIST1H3J	1	False	True	17
HIST1H3G	1	True	True	55
HIST1H3F	1	False	True	18
HIST1H3D	1	True	True	22
HIST1H3C	1	True	True	46
HIST1H3H	1	True	True	16
HIST1H3I	1	False	True	109
HIST2H2BE	1	False	False	260
OBSL1	1	False	False	29
HERC2	1	True	False	135
UBR4	1	False	False	614
WNK1	1	True	False	611
MDN1	1	False	False	624
FBXW11	1	False	False	244
DST	1	True	False	2
HIST3H3	1	False	True	0
TP53BP1	1	True	True	53
YWHAQ	1	False	False	1
SMC2	1	True	False	625
MCM7	1	False	False	625
E2F4	1	False	False	6
TPX2	1	True	False	624
CKAP5	1	False	False	625
BIRC5	1	False	False	625
FOXM1	1	True	False	193
RACGAP1	1	False	False	625
SMC4	1	True	False	623
CENPE	1	False	False	510
CCNB1	1	False	False	573

EXO1	1	False	False	30
PBK	1	False	False	0
UBB	1	True	False	24
CCNB2	1	True	False	0
CDC42	1	True	True	610
ADGRV1	1	True	False	0
TNRC6A	1	True	False	2
KMT2B	1	True	True	111
TNRC6C	1	False	False	4
MAPK11	1	False	False	0
TAF1L	1	True	False	0
USP34	1	False	False	5
NIPBL	1	True	False	229
PCLO	1	True	True	0
TNRC6B	1	False	False	14
BPTF	1	True	False	421
MAPK14	1	False	False	78
USP24	1	False	False	18
CTNNA1	1	True	True	34
MYCBP2	1	True	False	0
AGO3	1	False	False	5
CDH9	1	True	False	0
FANCM	1	True	False	158
HECW2	1	True	False	0
ALB	1	True	True	0
LYN	1	True	True	1
CEP135	1	False	False	208
BAG3	1	False	False	2
CRK	1	False	False	77
LRRK2	1	True	True	0
CALM1	1	False	False	7
GSK3B	1	False	True	10
USP7	1	True	False	495
NR3C1	1	True	False	4
SPI1	1	False	False	23
CTBP2	1	False	False	274
FOS	1	False	False	6
JUND	1	False	False	25

BATF	1	False	False	2
BMX	1	False	False	0
GTF2B	1	False	False	624
BDP1	1	False	False	579
POU2F2	1	True	False	5
PAX6	1	False	False	4
RFX5	1	False	False	57
PTK2B	1	True	False	2
TAF1	1	True	True	335
USF1	1	False	False	10
SUMO2	1	False	False	520
IRF3	1	False	False	0
HTT	1	False	False	39
FOXA2	1	True	False	33
HDAC5	1	False	False	3
KAT2A	1	False	False	21
IRF1	1	False	True	12
ATF3	1	False	False	2
CTCFL	1	False	False	0
NEK10	1	False	False	2
NEDD4	1	True	False	0
TFAP2A	1	False	False	21
PRKCA	1	False	False	9
CACNA1A	1	True	False	16
BTRC	1	False	False	0
CALM3	1	False	False	4
DLG4	1	False	False	29
SUMO1	1	False	False	7
SNTA1	1	False	False	0
UBQLN4	1	False	False	412
NCK1	1	False	False	1
ATN1	1	False	False	23
FN1	1	False	False	1
VCL	1	False	False	96
TLN1	1	False	False	619
DLG1	1	False	False	0
SOX10	1	False	True	174
ATF7IP	1	True	False	39

LRP2	1	True	False	11
TERF1	1	False	False	514
PRPF40A	1	False	False	621
STAT1	1	False	False	0
TGFBR1	1	True	True	49
HOXA5	1	False	False	14
PRKCD	1	False	False	1
MDFI	1	False	False	0

Table A.2: **List of NPCGs.** The table also depicts the number of models for which the gene was among the top 100 predictions (*Models top100* column), presence of the gene among the **CCGs** or oncoKB high confidence genes. Furthermore, the number of cell lines for which the gene was essential in **CRISPRi** loss-of-function screens from the Achilles project is depicted.

*Pathway Enrichment Analysis for **NPCGs***

ID	Pvalue	OddsRatio	ExpCount	Count	Size	Term
4110	6.58E-12	10.75809717	2.32994718	18	122	Cell cycle
4722	9.09E-11	9.741109358	2.36814303	17	124	Neurotrophin signaling pathway
5131	6.18E-10	14.88635184	1.12677773	12	59	Shigellosis
4114	8.30E-10	9.883850327	2.02438033	15	106	Oocyte meiosis
5220	4.99E-08	10.89945991	1.31775701	11	69	Chronic myeloid leukemia
4660	7.20E-08	8.171379265	2.02438033	13	106	T cell receptor signaling pathway
5200	2.86E-07	4.436399217	6.01584722	21	315	Pathways in cancer
4510	5.21E-07	5.390120238	3.68590004	16	193	Focal adhesion
4912	1.03E-06	7.76691953	1.75700935	11	92	GnRH signaling pathway
4380	2.67E-06	6.276718404	2.32994718	12	122	Osteoclast differentiation
4010	3.40E-06	4.373712494	4.75538399	17	249	MAPK signaling pathway
5100	3.47E-06	8.8625387	1.26046323	9	66	Bacterial invasion of epithelial cells
5130	5.74E-06	9.887338501	1.01219017	8	53	Pathogenic Escherichia coli infection
4062	7.05E-06	4.854166667	3.47582284	14	182	Chemokine signaling pathway
4664	7.26E-06	8.008403361	1.37505079	9	72	Fc epsilon RI signaling pathway
4520	8.15E-06	7.881617647	1.39414872	9	73	Adherens junction
4012	2.36E-05	6.802225755	1.585128	9	83	ErbB signaling pathway

5160	2.64E-05	5.336319638	2.44453474	11	128	Hepatitis C
4670	5.27E-05	5.407509158	2.17716375	10	114	Leukocyte transendothelial migration
4662	7.02E-05	6.711768851	1.41324665	8	74	B cell receptor signaling pathway
5214	0.0001	7.113239676	1.16497359	7	61	Glioma
4914	0.0001	5.895193798	1.585128	8	83	Progesterone-mediated oocyte maturation
4330	0.00016	8.594497608	0.84030882	6	44	Notch signaling pathway
5140	0.00036	5.989224138	1.35595286	7	71	Leishmaniasis
5221	0.00056	6.649814471	1.05038602	6	55	Acute myeloid leukemia
4620	0.00057	4.78867543	1.90979277	8	100	Toll-like receptor signaling pathway
4910	0.00071	4.154117647	2.46363267	9	129	Insulin signaling pathway
5215	0.00133	4.715339861	1.68061764	7	88	Prostate cancer
4810	0.00152	3.200050201	3.87687932	11	203	Regulation of actin cytoskeleton
4370	0.00174	5.241202346	1.29865908	6	68	VEGF signaling pathway
5120	0.00174	5.241202346	1.29865908	6	68	Epithelial cell signaling in Helicobacter pylori infection
5212	0.00188	5.156926407	1.31775701	6	69	Pancreatic cancer
5213	0.00267	5.84025403	0.97399431	5	51	Endometrial cancer
5142	0.00312	4.008590442	1.94798862	7	102	Chagas disease (American trypanosomiasis)
4621	0.00404	5.262172285	1.06948395	5	56	NOD-like receptor signaling pathway
5210	0.00584	4.787319422	1.16497359	5	61	Colorectal cancer
4666	0.00748	3.804545455	1.73791142	6	91	Fc gamma R-mediated phagocytosis
4720	0.00814	4.390311291	1.26046323	5	66	Long-term potentiation
4622	0.00814	4.390311291	1.26046323	5	66	RIG-I-like receptor signaling pathway
4971	0.00814	4.390311291	1.26046323	5	66	Gastric acid secretion
4115	0.00814	4.390311291	1.26046323	5	66	p53 signaling pathway
4320	0.00905	7.925274725	0.43925234	3	23	Dorso-ventral axis formation
4650	0.00904	3.239709205	2.36814303	7	124	Natural killer cell mediated cytotoxicity
5145	0.00946	3.21157218	2.38724096	7	125	Toxoplasmosis

5211	0.00979	4.181882022	1.31775701	5	69	Renal cell carcinoma
4530	0.01208	3.052280311	2.50182852	7	131	Tight junction
5146	0.01337	3.325445173	1.96708655	6	103	Amoebiasis
4270	0.01337	3.325445173	1.96708655	6	103	Vascular smooth muscle contraction
4340	0.01372	4.723950617	0.93579846	4	49	Hedgehog signaling pathway
5110	0.01678	4.425925926	0.99309224	4	52	Vibrio cholerae infection
5223	0.01790	4.334693878	1.01219017	4	53	Non-small cell lung cancer
4350	0.01869	3.51271437	1.54693214	5	81	TGF-beta signaling pathway
5222	0.02154	3.377186744	1.60422592	5	84	Small cell lung cancer
4540	0.02255	3.334269663	1.62332385	5	85	Gap junction
4360	0.02549	2.844931617	2.27265339	6	119	Axon guidance
3022	0.02632	5.101382488	0.64932954	3	34	Basal transcription factors
4730	0.03153	3.592467043	1.20316944	4	63	Long-term depression
4916	0.0317	3.026046987	1.77610727	5	93	Melanogenesis
5016	0.04284	2.302729004	3.2466477	7	170	Huntington's disease

Table A.3: **KEGG pathway enrichment for NPCGs.**

A.10 BICLUSTERING OF LRP FEATURE CONTRIBUTIONS

Biclustering of all Genes

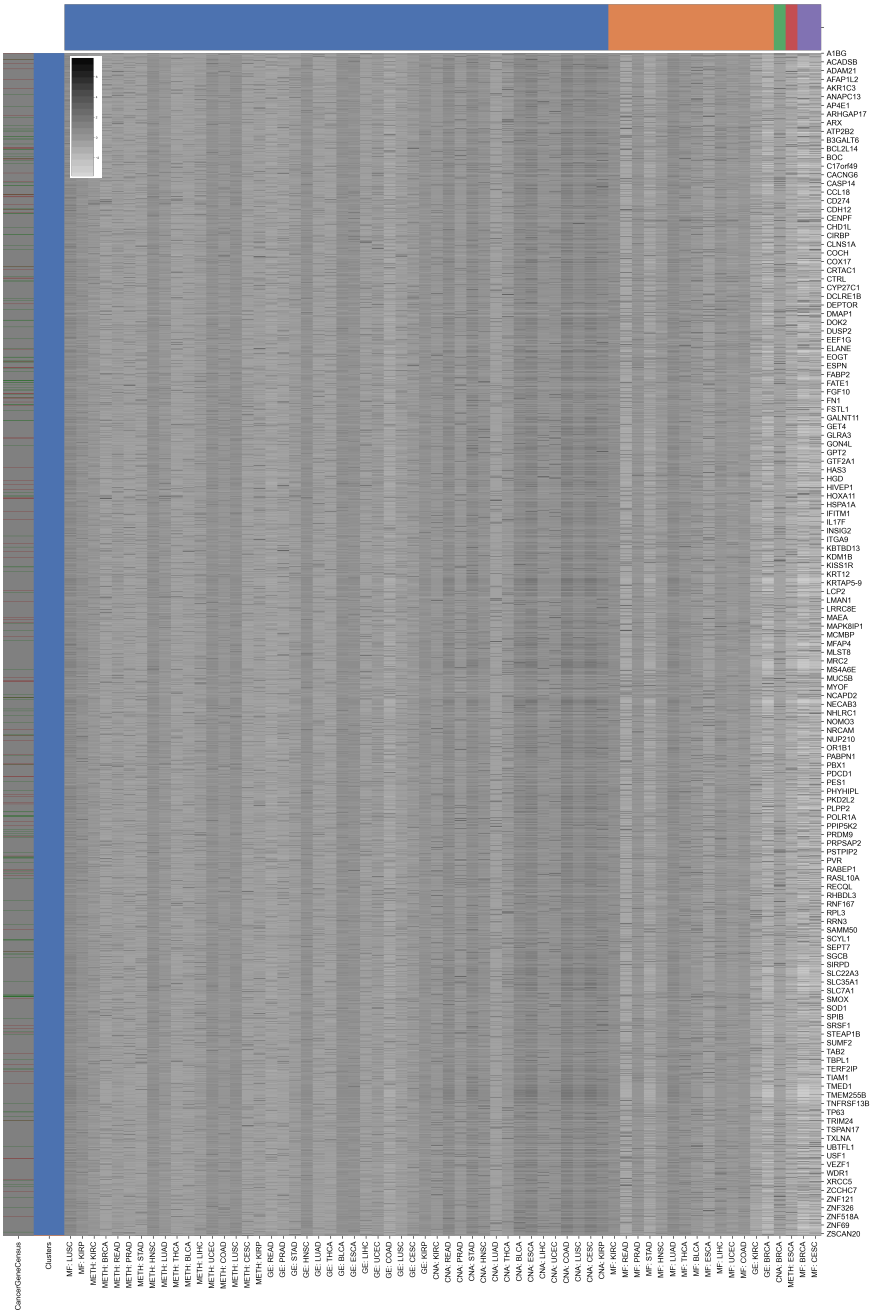


Figure A.13: **Biclustering analysis for all genes in the CPDB PPI network.** The rows correspond to genes and the columns to data types. the same processing steps were used as for the biclustering analysis of the top 1000 genes.

Gene Ontology Enrichment Analysis for Biclusters

A.11 MODULE DISCOVERY IN THE **CPDB PPI** NETWORK

*Correlation between Edge Betweenness & **LRP** Edge Weight*

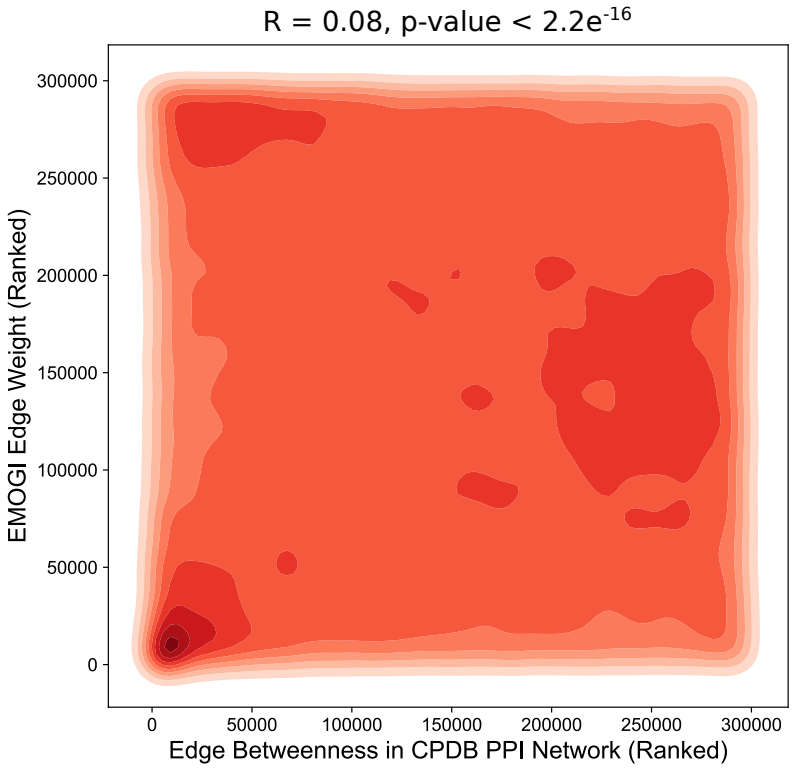


Figure A.14: **Ranked correlation between LRP edge weights and edge betweenness.** The contour plot depicts ranked weights of edges according to the summed **LRP** interaction contributions (E^{I}_{total}, see Section 5.8 for details) in comparison with edge betweenness. The latter is a global metric of edge importance and measures the fraction of shortest paths between nodes that pass through the edge. Spearman correlation is shown on top with a correlation coefficient of 0.08.

*Gene Ontology Enrichment Analysis for the largest **SCCs***

Name	Ratio Study	Ratio Pop.	Pvalue	Depth	Count	Pval. Corr.
extracellular matrix organization	19/148	253/20913	2E-14	5	19	2.4E-10
cell-cell adhesion	10/148	143/20913	7.4E-08	3	10	0.00046

cerebral cortex cell migration	4/148	8/20913	1.6E-07	7	4	0.00064
platelet degranulation	9/148	124/20913	2.5E-07	8	9	0.00064
cellular response to amyloid-beta	6/148	38/20913	2.6E-07	7	6	0.00064
cellular protein metabolic process	11/148	225/20913	6.2E-07	5	11	0.00126
astrocyte activation involved in immune response	3/148	4/20913	1.4E-06	5	3	0.00212
positive regulation of amyloid fibril formation	3/148	4/20913	1.4E-06	7	3	0.00212
response to amyloid-beta	4/148	14/20913	2.3E-06	6	4	0.00311
ephrin receptor signaling pathway	7/148	86/20913	2.6E-06	8	7	0.00319
cellular response to indole-3-methanol	3/148	5/20913	3.4E-06	6	3	0.00384
positive regulation of gene expression	13/148	392/20913	4.5E-06	6	13	0.0046
transforming growth factor beta receptor signaling pathway	7/148	95/20913	5.1E-06	8	7	0.00478
response to drug	11/148	284/20913	5.9E-06	3	11	0.00516
receptor-mediated endocytosis	7/148	100/20913	7.1E-06	6	7	0.00583
positive regulation of amyloid-beta formation	4/148	19/20913	8.6E-06	8	4	0.00659
collagen catabolic process	5/148	43/20913	1.3E-05	3	5	0.0091
neuron migration	7/148	110/20913	1.3E-05	5	7	0.0091
bone mineralization	5/148	45/20913	1.6E-05	5	5	0.01032
MAPK cascade	10/148	260/20913	1.7E-05	9	10	0.01032
entry of bacterium into host cell	3/148	8/20913	1.9E-05	6	3	0.01108

response to hypoxia	8/148	163/20913	2.1E-05	5	8	0.01196
positive regulation of phosphorylation	4/148	26/20913	3.2E-05	8	4	0.01702
wound healing	6/148	89/20913	4.1E-05	4	6	0.0208
protein processing	5/148	56/20913	4.7E-05	6	5	0.02326
smooth endoplasmic reticulum calcium ion homeostasis	2/148	2/20913	5E-05	11	2	0.02348
amyloid precursor protein metabolic process	3/148	11/20913	5.5E-05	5	3	0.02499
regulation of neuron projection development	4/148	31/20913	6.5E-05	8	4	0.02863
learning or memory	5/148	61/20913	7.2E-05	5	5	0.03036
extracellular matrix disassembly	5/148	64/20913	9E-05	6	5	0.03698
positive regulation of NIK/NF-kappaB signaling	5/148	65/20913	9.7E-05	8	5	0.03855
extracellular region	54/148	1793/20913	3.9E-21	2	54	6.8E-18
collagen-containing extracellular matrix	23/148	374/20913	2.2E-15	3	23	1.4E-12
extracellular space	42/148	1472/20913	2.3E-15	2	42	1.4E-12
extracellular exosome	43/148	2093/20913	6.9E-11	6	43	3E-08
endoplasmic reticulum lumen	15/148	299/20913	3.6E-09	5	15	1.3E-06
extracellular matrix	13/148	235/20913	1.3E-08	2	13	3.8E-06
plasma membrane	61/148	4401/20913	2.4E-08	3	61	5.9E-06
perinuclear region of cytoplasm	20/148	693/20913	9.9E-08	2	20	2.2E-05
endoplasmic reticulum	23/148	1016/20913	7.6E-07	5	23	0.00014

lamellipodium	10/148	184/20913	7.8E-07	4	10	0.00014
Wnt signalosome	4/148	12/20913	1.1E-06	2	4	0.00018
Golgi apparatus	22/148	992/20913	1.9E-06	5	22	0.00028
endosome lumen	4/148	18/20913	6.8E-06	5	4	0.00092
platelet alpha granule lumen	6/148	67/20913	7.9E-06	7	6	0.00099
Golgi lumen	7/148	103/20913	8.7E-06	5	7	0.00101
cytoplasm	56/148	4626/20913	1.5E-05	2	56	0.00161
receptor complex	9/148	207/20913	1.7E-05	2	9	0.00177
cell surface	15/148	591/20913	2E-05	2	15	0.00197
cytoskeleton	12/148	403/20913	3.1E-05	5	12	0.00284
focal adhesion	12/148	415/20913	4.1E-05	5	12	0.00359
cell-cell junction	8/148	184/20913	5.1E-05	4	8	0.00425
clathrin-coated pit	5/148	59/20913	6.1E-05	4	5	0.00477
cell junction	8/148	190/20913	6.4E-05	2	8	0.00477
catenin complex	4/148	31/20913	6.5E-05	4	4	0.00477
axon	10/148	309/20913	7.2E-05	5	10	0.00492
beta-catenin destruction complex	3/148	12/20913	7.3E-05	2	3	0.00492
lateral plasma membrane	5/148	65/20913	9.7E-05	2	5	0.00632
adherens junction	7/148	157/20913	0.00013	5	7	0.00812
blood microparticle	6/148	112/20913	0.00015	2	6	0.00868
node of Ranvier	3/148	15/20913	0.00015	2	3	0.00868
apical part of cell	5/148	74/20913	0.00018	2	5	0.01021
protein-containing complex	14/148	644/20913	0.0002	1	14	0.01087

endosome	9/148	294/20913	0.00025	7	9	0.01344
membrane raft	8/148	235/20913	0.00028	5	8	0.01422
synaptic vesicle	6/148	131/20913	0.00034	9	6	0.01719
neuron projection	9/148	338/20913	0.00069	4	9	0.03289
spindle midzone	3/148	25/20913	0.00071	2	3	0.03289
desmosome	3/148	25/20913	0.00071	5	3	0.03289
gamma-secretase complex	2/148	6/20913	0.00073	4	2	0.03294
smooth endoplasmic reticulum	3/148	27/20913	0.0009	6	3	0.03837
nuclear outer membrane	3/148	27/20913	0.0009	6	3	0.03837
glutamatergic synapse	9/148	353/20913	0.00094	4	9	0.03937
death-inducing signaling complex	2/148	7/20913	0.00102	4	2	0.04163
calcium ion binding	28/148	698/20913	7.7E-14	5	28	3.2E-10
heparin binding	12/148	168/20913	2.8E-09	4	12	5.9E-06
protein binding	119/148	12001/20913	4.4E-09	2	119	6E-06
low-density lipoprotein particle receptor binding	6/148	21/20913	5.6E-09	5	6	6E-06
protease binding	9/148	106/20913	6.4E-08	4	9	4.9E-05
integrin binding	10/148	142/20913	6.9E-08	4	10	4.9E-05
growth factor receptor binding	4/148	10/20913	4.9E-07	4	4	0.0003
heparan sulfate proteoglycan binding	4/148	16/20913	4.1E-06	4	4	0.00217
extracellular matrix structural constituent	8/148	137/20913	6E-06	2	8	0.00284
signaling receptor binding	12/148	353/20913	8.3E-06	3	12	0.00351
amyloid-beta binding	6/148	77/20913	1.8E-05	4	6	0.00678
transforming growth factor beta binding	4/148	23/20913	1.9E-05	4	4	0.00678

transforming growth factor beta-activated receptor activity	3/148	9/20913	2.8E-05	8	3	0.00856
cysteine-type endopeptidase activity involved in execution phase of apoptosis	3/148	9/20913	2.8E-05	7	3	0.00856
peptidase activator activity	3/148	10/20913	4E-05	4	3	0.01065
cysteine-type endopeptidase activity involved in apoptotic process	3/148	10/20913	4E-05	6	3	0.01065
guanosine-diphosphatase activity	2/148	2/20913	5E-05	7	2	0.01172
triglyceride binding	2/148	2/20913	5E-05	3	2	0.01172
enzyme binding	11/148	360/20913	5.3E-05	3	11	0.01177
phospholipase A1 activity	3/148	12/20913	7.3E-05	6	3	0.01545
cadherin binding	10/148	315/20913	8.5E-05	4	10	0.01715
collagen binding	5/148	67/20913	0.00011	3	5	0.02024
unfolded protein binding	6/148	107/20913	0.00011	3	6	0.02024
identical protein binding	24/148	1466/20913	0.00012	3	24	0.02024
cargo receptor activity	3/148	14/20913	0.00012	1	3	0.02024
apolipoprotein binding	3/148	16/20913	0.00018	3	3	0.02964
ubiquitin protein ligase binding	9/148	294/20913	0.00025	5	9	0.03799
endopeptidase activity	5/148	81/20913	0.00028	4	5	0.03799
tau protein binding	4/148	45/20913	0.00029	4	4	0.03799
uridine-diphosphatase activity	2/148	4/20913	0.0003	7	2	0.03799
CD8 receptor binding	2/148	4/20913	0.0003	4	2	0.03799
choline binding	2/148	4/20913	0.0003	4	2	0.03799
calcium ion sensor activity	2/148	4/20913	0.0003	6	2	0.03799
beta-catenin binding	5/148	85/20913	0.00035	3	5	0.04304

protein binding	phosphatase	5/148	86/20913	0.00036	5	5	0.04414

Table A.4: **Gene Ontology Enrichment Analysis (GOEA) for the largest SCC.**

*Pathway Enrichment Analysis for the largest **SCCs***

KEGGID	Pvalue	Odds	Exp. Count	Count	Size	Term
4510	1.57E-05	5	3	12	193	Focal adhesion
5200	3.05E-05	4	5	15	315	Pathways in cancer
5213	7.85E-05	10	0.735677	6	51	Endometrial cancer
4350	0.000143	7	1	7	81	TGF-beta signaling pathway
4512	0.000943	6	1	6	80	ECM-receptor interaction
5010	0.001529	4	2	8	154	Alzheimer's disease
5210	0.001722	6	0.879927	5	61	Colorectal cancer
5218	0.002615	6	0.966477	5	67	Melanoma
5219	0.002692	8	0.591426	4	41	Bladder cancer
4610	0.002976	6	0.995327	5	69	Complement and coagulation cascades
4520	0.003805	5	1	5	73	Adherens junction
5130	0.006837	6	0.764527	4	53	Pathogenic Escherichia coli infection
4810	0.008312	3	3	8	203	Regulation of actin cytoskeleton
5215	0.008404	4	1	5	88	Prostate cancer
4310	0.014481	3	2	6	139	Wnt signaling pathway
5100	0.014631	5	0.952052	4	66	Bacterial invasion of epithelial cells
4660	0.017819	4	2	5	106	T cell receptor signaling pathway
5416	0.017831	4	1	4	70	Viral myocarditis
4360	0.027835	3	2	5	119	Axon guidance
4012	0.031102	4	1	4	83	ErbB signaling pathway
4210	0.032311	4	1	4	84	Apoptosis
4340	0.033029	5	0.706826	3	49	Hedgehog signaling pathway
5217	0.034783	5	0.721252	3	50	Basal cell carcinoma
4320	0.042708	7	0.331776	2	23	Dorso-ventral axis formation
4916	0.044432	3	1	4	93	Melanogenesis
4744	0.049721	6	0.360626	2	25	Phototransduction
5142	0.058791	3	1	4	102	Chagas disease (American trypanosomiasis)
5146	0.060523	3	1	4	103	Amoebiasis
5216	0.064871	5	0.418326	2	29	Thyroid cancer
4141	0.065734	2	2	5	151	Protein processing in endoplasmic reticulum
4010	0.066048	2	4	7	249	MAPK signaling pathway
4742	0.068874	5	0.432751	2	30	Taste transduction
4115	0.069109	3	0.952052	3	66	p53 signaling pathway
61	0.070102	17	0.072125	1	5	Fatty acid biosynthesis
4670	0.08135	3	2	4	114	Leukocyte transendothelial migration

Table A.5: **KEGG Pathway enrichment for the largest SCC.**

B

CONTRIBUTION TO TRIPEPSVM

During the time of my PhD, I was involved in another project other than the prediction of cancer-associated genes. Together with Annkatrin Bressin and Annalisa Marsico, and in collaboration with the lab of Benedikt Beckmann, I developed *TriPepSVM*, a machine-learning method to predict RNA-binding proteins in bacteria and humans.
In Section 2.1.2 we saw that gene regulation is complex and occurs at several different levels. One of such layers is the regulation of RNA after transcription, termed *post-transcriptional regulation*. An important aspect of post-transcriptional regulation are RNA-binding proteins (RBPs) which form ribonucleoprotein complexes (RNPs) by dynamic, transient interactions, and control different steps in RNA metabolism, such as RNA stability, degradation, splicing and polyadenylation. Numerous diseases have already been linked to defects in RBP expression and function, among them cancer diseases [284–286]. In the project, a support-vector machine (SVM) was trained on the genomic sequences of known RNA-binding proteins (positive set) and non-RNA binders (negative set) for different species. For that, TriPepSVM makes use of a string kernel [287] that allows to use strings in the SVM framework. Briefly, string kernels compute all sub-strings of length k for a given string (or RNA sequence in this case) and arrange them in a vector representation where every entry corresponds to a sub-string. The values in the vector indicate the number of times that the sub-string was present in the sequence. String kernels in general correspond to a linear transformation, making TriPepSVM an interpretable machine-learning method (as demonstrated in Section 4.6).
Interestingly, this simple approach that only relied on the frequencies of k-mers (sub-strings of length k) outperformed or performed similar to other methods that use much more information to distinguish RNA binders from non-RNA-binding proteins, even when more training data was available (e.g. for human). Leveraging feature interpretation of the most important k-mers, we found an enrichment with intrinsically disordered regions (IDRs). Interestingly, this fits very well with the observation that IDRs are important in liquid-liquid phase transitions [288].
Finally, our collaborators could experimentally validate three out of four newly predicted RNA-binding proteins, in line with an accuracy of around 80%.
My role in the project was to continue the development and benchmarking of TriPepSVM and to set up and maintain a github repository for the project. The project along with all analyses is publically available at `https://github.com/marsicoLab/TriPepSVM`.

BIBLIOGRAPHY

1. Alberts, B. *Molecular biology of the cell* ISBN: 0815344643 (1983).
2. Gnirke, A. *et al.* Solution hybrid selection with ultra-long oligonucleotides for massively parallel targeted sequencing. *Nature Biotechnology* **27,** 182–189. ISSN: 10870156 (Feb. 2009).
3. Hekselman, I. & Yeger-Lotem, E. *Mechanisms of tissue and cell-type specificity in heritable traits and diseases* Mar. 2020.
4. Gasperini, M., Tome, J. M. & Shendure, J. *Towards a comprehensive catalogue of validated and target-linked human enhancers* Jan. 2020.
5. Wingender, E. *et al.* TFClass: A classification of human transcription factors and their rodent orthologs. *Nucleic Acids Research* **43,** D97–D102. ISSN: 13624962 (2015).
6. Crick, F. Central dogma of molecular biology. *Nature* **227,** 561–563. ISSN: 00280836. https://pubmed.ncbi.nlm.nih.gov/4913914/ (1970).
7. Sonawane, A. R. *et al.* Understanding Tissue-Specific Gene Regulation. *Cell Reports* **21,** 1077–1088. ISSN: 22111247 (Oct. 2017).
8. Klemm, S. L., Shipony, Z. & Greenleaf, W. J. *Chromatin accessibility and the regulatory epigenome* Apr. 2019. www.nature.com/nrg.
9. Filipowicz, W., Bhattacharyya, S. N. & Sonenberg, N. *Mechanisms of post-transcriptional regulation by microRNAs: Are the answers in sight?* Feb. 2008. www.nature.com/reviews/genetics.
10. Fernandez, A. F. *et al.* A DNA methylation fingerprint of 1628 human samples. *Genome Research* **22,** 407–419. ISSN: 10889051 (Feb. 2012).
11. Tost, J. *DNA methylation: an introduction to the biology and the disease-associated changes of a promising biomarker.* 2009. https://pubmed.ncbi.nlm.nih.gov/18987802/.
12. Baylin, S. B. *DNA methylation and gene silencing in cancer* Dec. 2005. http://www.nature.com/articles/ncponc0354.
13. Choy, M. K. *et al.* Genome-wide conserved consensus transcription factor binding motifs are hyper-methylated. *BMC Genomics* **11.** ISSN: 14712164. https://pubmed.ncbi.nlm.nih.gov/20875111/ (Sept. 2010).
14. Greenberg, M. V. & Bourc'his, D. The diverse roles of DNA methylation in mammalian development and disease. *Nature Reviews Molecular Cell Biology* **20,** 590–607. ISSN: 14710080. www.nature.com/nrm (Oct. 2019).
15. Du, J. *et al.* DNA methylation pathways and their crosstalk with histone methylation. *Nature Reviews Molecular Cell Biology* **16,** 519–532. ISSN: 14710080 (Aug. 2015).

16. Young, R. A. *RNA Polymerase II* June 1991. http://www.annualreviews.org/doi/10.1146/annurev.bi.60.070191.003353.

17. Ashburner, M. *et al. Gene ontology: Tool for the unification of biology* May 2000.

18. Boveri, T. The origin of malignant tumours. *Baltimore, MD: Williams & Wilkins.* ISSN: 0021-9533. http://jcs.biologists.org/cgi/doi/10.1242/jcs.024562 (1929).

19. Garraway, L. A. & Lander, E. S. *Lessons from the cancer genome* 2013.

20. Beroukhim, R. *et al.* The landscape of somatic copy-number alteration across human cancers. *Nature* **463,** 899–905. ISSN: 1476-4687. http://www.ncbi.nlm.nih.gov/pubmed/20164920%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2826709 (Feb. 2010).

21. Zack, T. I. *et al.* Pan-cancer patterns of somatic copy number alteration. *Nature Genetics* **45,** 1134–1140. ISSN: 10614036. arXiv: 15334406. https://www.nature.com/articles/ng.2760.pdf (2013).

22. Stratton, M. R., Campbell, P. J. & Futreal, P. A. *The cancer genome* Apr. 2009. /pmc/articles/PMC2821689/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2821689/.

23. Hanahan, D. & Weinberg, R. A. Hallmarks of cancer: The next generation. *Cell* **144,** 646–674. ISSN: 00928674. http://www.ncbi.nlm.nih.gov/pubmed/21376230 (Mar. 2011).

24. Stratton, M. R. *Exploring the genomes of cancer cells: Progress and promise* Mar. 2011.

25. Vogelstein, B. *et al.* Cancer genome landscapes. *Science* **340,** 1546–1558. ISSN: 10959203. http://science.sciencemag.org/content/sci/339/6127/1546.full.pdf%20http://dx.doi.org/10.1126/science.1235122 (Mar. 2013).

26. Gerstung, M. *et al.* The evolutionary history of 2,658 cancers. *Nature* **578,** 122–128. ISSN: 14764687. https://doi.org/10.1038/s41586-019-1907-7 (Feb. 2020).

27. Khurana, E. *et al.* Role of non-coding sequence variants in cancer. *Nat Rev Genet* **17,** 93–108. ISSN: 1471-0056. http://dx.doi.org/10.1038/nrg.2015.17%7B%5C%%7D5Cnhttp://10.1038/nrg.2015.17 (2016).

28. Campbell, P. J. *et al.* Pan-cancer analysis of whole genomes. *Nature* **578,** 82–93. ISSN: 14764687 (Feb. 2020).

29. Lawrence, M. S. *et al.* Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature* **505,** 495–501. ISSN: 00280836 (2014).

30. Cancer Genome Atlas Research Network, J. N. *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nature genetics* **45,** 1113–20. ISSN: 1546-1718. http://www.ncbi.nlm.nih.gov/pubmed/24071849%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3919969 (Oct. 2013).

31. Tokheim, C. J. *et al.* Evaluating the evaluation of cancer driver genes. *Proceedings of the National Academy of Sciences of the United States of America* **113,** 14330–14335. ISSN: 10916490. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5167163/%20www.pnas.org/cgi/doi/10.1073/pnas.1616440113 (2016).

32. Greaves, M. & Maley, C. C. *Clonal evolution in cancer* Jan. 2012. https://www.nature.com/articles/nature10762.

33. Yates, L. R. & Campbell, P. J. *Evolution of the cancer genome* Nov. 2012. www.nature.com/reviews/genetics.

34. Goutsouliak, K. *et al. Towards personalized treatment for early stage HER2-positive breast cancer* Apr. 2020. https://www.nature.com/articles/s41571-019-0299-9.

35. Pophali, P. A. & Patnaik, M. M. *The role of new tyrosine kinase inhibitors in chronic myeloid leukemia* Jan. 2016. /pmc/articles/PMC4742366/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4742366/.

36. Cancer.Net. *Understanding Targeted Therapy | Cancer.Net* 2016. https://www.cancer.net/navigating-cancer-care/how-cancer-treated/personalized-and-targeted-therapies/understanding-targeted-therapy%20https://www.cancer.net/navigating-cancer-care/how-cancer-treated/personalized-and-targeted-therapies/understanding-targeted (2020).

37. Bradner, J. E., Hnisz, D. & Young, R. A. *Transcriptional Addiction in Cancer* 2017.

38. Stehelin, D. *et al.* DNA related to the transforming gene(s) of avian sarcoma viruses is present in normal avian DNA. *Nature* **260,** 170–173. ISSN: 00280836 (1976).

39. Tabin, C. J. *et al.* Mechanism of activation of a human oncogene. *Nature* **300,** 143–149. ISSN: 00280836 (1982).

40. Lawrence, M. S. *et al.* Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature* **499,** 214–218. ISSN: 1476-4687. http://www.ncbi.nlm.nih.gov/pubmed/23770567%7B%5C%%7D0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3919509 (2013).

41. Leiserson, M. D. *et al.* Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature Genetics* **47,** 106–114. ISSN: 15461718. arXiv: 15334406bradner201. http://www.nature.com/articles/ng.3168%20https://www.nature.com/ng/journal/v47/n2/pdf/ng.3168.pdf%20http://www.nature.com/doifinder/10.1038/ng.3168 (2015).

42. Reyna, M. A., Leiserson, M. D. & Raphael, B. J. *Hierarchical HotNet: Identifying hierarchies of altered subnetworks* in *Bioinformatics* **34** (Oxford University Press, Sept. 2018), i972–i980. https://academic.oup.com/bioinformatics/article/34/17/i972/5093236.

43. Han, Y. *et al.* DriverML: a machine learning algorithm for identifying driver genes in cancer sequencing studies. *Nucleic Acids Research* **47,** e45–e45. ISSN: 0305-1048. https://academic.oup.com/nar/article/47/8/e45/5324448 (May 2019).

44. Collier, O., Stoven, V. & Vert, J.-P. LOTUS: A single- and multitask machine learning algorithm for the prediction of cancer driver genes. *PLOS Computational Biology* **15,** e1007381. ISSN: 15537358 (Sept. 2019).

45. Smith, Z. D. *et al.* Epigenetic restriction of extraembryonic lineages mirrors the somatic transition to cancer. *Nature* **549,** 543–547. ISSN: 14764687 (Sept. 2017).

46. Patani, H. *et al.* Transition to naïve human pluripotency mirrors pan-cancer DNA hypermethylation. *Nature Communications* **11,** 1–17. ISSN: 20411723 (Dec. 2020).

47. Jamieson, C. Bad blood promotes tumour progression. *Nature* **549,** 465–466. ISSN: 14764687 (Sept. 2017).

48. Chen, H. & He, X. The convergent cancer evolution toward a single cellular destination. *Molecular Biology and Evolution* **33,** 4–12. ISSN: 15371719. `https://pubmed.ncbi.nlm.nih.gov/26464125/` (Jan. 2016).

49. Silverbush, D. *et al.* Simultaneous Integration of Multi-omics Data Improves the Identification of Cancer Driver Modules. *Cell Systems* **8,** 456–466.e5. ISSN: 24054720. `http://www.ncbi.nlm.nih.gov/pubmed/31103572` (May 2019).

50. Reyna, M. A. *et al.* Pathway and network analysis of more than 2500 whole cancer genomes. *Nature communications* **11,** 729. ISSN: 2041-1723. `http://www.ncbi.nlm.nih.gov/pubmed/32024854` (Feb. 2020).

51. Bell, C. C. & Gilan, O. *Principles and mechanisms of non-genetic resistance in cancer* Feb. 2019.

52. Rheinbay, E. *et al.* Analyses of non-coding somatic drivers in 2,658 cancer whole genomes. *Nature* **578,** 102–111. ISSN: 14764687 (Feb. 2020).

53. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research* **26,** 990–999. ISSN: 15495469 (July 2016).

54. Stark, R., Grzelak, M. & Hadfield, J. *RNA sequencing: the teenage years* Nov. 2019.

55. Wang, Q. *et al.* Data Descriptor: Unifying cancer and normal RNA sequencing data from different sources. *Scientific Data* **5,** 1–8. ISSN: 20524463. `http://www.nature.com/articles/sdata201861` (2018).

56. Silberstein, G. B., Dressler, G. R. & Van Horn, K. Expression of the PAX2 oncogene in human breast cancer and its role in progesterone-dependent mammary growth. *Oncogene* **21,** 1009–1016. ISSN: 09509232. `https://pubmed.ncbi.nlm.nih.gov/11850818/` (2002).

57. Oh, D. Y. *et al.* HER2 as a novel therapeutic target for cervical cancer. *Oncotarget* **6,** 36219–36230. ISSN: 19492553 (2015).

58. Hagemann, N. *et al.* The serologically defined colon cancer antigen-3 interacts with the protein tyrosine phosphatase PTPN13 and is involved in the regulation of cytokinesis. *Oncogene* **32,** 4602–4613. ISSN: 09509232. `www.nature.com/onc` (Sept. 2013).

59. Chen, H. Z., Tsai, S. Y. & Leone, G. *Emerging roles of E2Fs in cancer: An exit from cell cycle control* Nov. 2009. http://www.ncbi.nlm.nih.gov/pubmed/19851314%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3616489.

60. Dang, C. V. MYC on the path to cancer. *Cell* **149,** 22–35. ISSN: 00928674. http://www.ncbi.nlm.nih.gov/pubmed/22464321%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3345192 (Mar. 2012).

61. Baylin, S. B. & Jones, P. A. Epigenetic determinants of cancer. *Cold Spring Harbor Perspectives in Biology* **8.** ISSN: 19430264 (Sept. 2016).

62. Klutstein, M. *et al. DNA methylation in cancer and aging* June 2016. http://www.ncbi.nlm.nih.gov/pubmed/27256564.

63. Gruber, M. *et al.* Growth dynamics in naturally progressing chronic lymphocytic leukaemia. *Nature* **570,** 474–479. ISSN: 0028-0836. http://www.nature.com/articles/s41586-019-1252-x (June 2019).

64. Beggs, A. D. *et al.* Whole-genome methylation analysis of benign and malignant colorectal tumours. *Journal of Pathology* **229,** 697–704. ISSN: 00223417. /pmc/articles/PMC3619233/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3619233/ (Apr. 2013).

65. Wei, J. *et al.* Discovery and validation of hypermethylated markers for colorectal cancer. *Disease Markers* **2016.** ISSN: 18758630. /pmc/articles/PMC4963574/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4963574/ (2016).

66. Bormann, F. *et al.* Cell-of-Origin DNA Methylation Signatures Are Maintained during Colorectal Carcinogenesis. *Cell Reports* **23,** 3407–3418. ISSN: 22111247. http://www.ncbi.nlm.nih.gov/pubmed/29898408 (June 2018).

67. Jones, P. A. & Baylin, S. B. *The Epigenomics of Cancer* Feb. 2007. http://www.cell.com/article/S0092867407001274/fulltext%20http://www.cell.com/article/S0092867407001274/abstract%20https://www.cell.com/cell/abstract/S0092-8674(07)00127-4.

68. Vandin, F. *et al. Discovery of mutated subnetworks associated with clinical data in cancer* in *Pacific Symposium on Biocomputing* (WORLD SCIENTIFIC, Dec. 2012), 55–66. ISBN: 978-981-4596-37-4. http://www.worldscientific.com/doi/abs/10.1142/9789814366496%7B%5C_%7D0006.

69. Cowen, L. *et al.* Network propagation: A universal amplifier of genetic associations. *Nature Reviews Genetics* **18,** 551–562. ISSN: 14710064. https://www.nature.com/nrg/journal/vaop/ncurrent/pdf/nrg.2017.38.pdf%20http://www.nature.com.remote.library.osaka-u.ac.jp/nrg/journal/vaop/ncurrent/pdf/nrg.2017.38.pdf (2017).

70. Barel, G. & Herwig, R. NetCore: A network propagation approach using node coreness. *Nucleic Acids Research* **48,** 98. ISSN: 13624962. https://academic.oup.com/nar/article/48/17/e98/5879427 (Sept. 2020).

71. Rappoport, N. & Shamir, R. *Multi-omic and multi-view clustering algorithms: Review and cancer benchmark* 2018.

72. John Naisbitt. *Megatrends: Ten New Directions Transforming Our Lives* ISBN: 978-0446512510 (Sept. 1982).

73. Sanger, F., Nicklen, S. & Coulson, A. R. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America* **74,** 5463–5467. ISSN: 00278424. /pmc/articles/PMC431765/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC431765/ (1977).

74. Schena, M. *et al.* Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270,** 467–470. ISSN: 00368075. https://pubmed.ncbi.nlm.nih.gov/7569999/ (1995).

75. Heather, J. M. & Chain, B. *The sequence of sequencers: The history of sequencing DNA* Jan. 2016.

76. Goodwin, S., McPherson, J. D. & McCombie, W. R. *Coming of age: Ten years of next-generation sequencing technologies* June 2016.

77. Metzker, M. L. *Sequencing technologies the next generation* Jan. 2010.

78. Ng, S. B. *et al.* Exome sequencing identifies the cause of a mendelian disorder. *Nature Genetics* **42,** 30–35. ISSN: 10614036 (Jan. 2010).

79. Meyerson, M., Gabriel, S. & Getz, G. *Advances in understanding cancer genomes through second-generation sequencing* Oct. 2010. http://www.ncbi.nlm.nih.gov/pubmed/20847746.

80. Goya, R. *et al.* SNVMix: Predicting single nucleotide variants from next-generation sequencing of tumors. *Bioinformatics* **26,** 730–736. ISSN: 13674803. https://pubmed.ncbi.nlm.nih.gov/20130035/ (Feb. 2010).

81. Dees, N. D. *et al.* MuSiC: Identifying mutational significance in cancer genomes. *Genome Research* **22,** 1589–1598. ISSN: 10889051. /pmc/articles/PMC3409272/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3409272/ (Aug. 2012).

82. Campbell, P. J. *et al.* Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nature Genetics* **40,** 722–729. ISSN: 10614036. http://www.ncbi.nlm.nih.gov/pubmed/18438408%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2705838 (June 2008).

83. Chiang, D. Y. *et al.* High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nature Methods* **6,** 99–103. ISSN: 15487091. http://www.ncbi.nlm.nih.gov/pubmed/19043412%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2630795 (Jan. 2009).

84. Mermel, C. H. *et al.* GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biology* **12,** R41. ISSN: 1465-6906. http://genomebiology.biomedcentral.com/articles/10.1186/gb-2011-12-4-r41 (Apr. 2011).

85. Ladd-Acosta, C. *et al.* DNA methylation signatures within the human brain. *American Journal of Human Genetics* **81,** 1304–1315. ISSN: 00029297. https://pubmed.ncbi.nlm.nih.gov/17999367/ (2007).

86. Bock, C. *Analysing and interpreting DNA methylation data.* Sept. 2012.

87. Bibikova, M. *et al.* High density DNA methylation array with single CpG site resolution. *Genomics* **98,** 288–295. ISSN: 08887543. https://pubmed.ncbi.nlm.nih.gov/21839163/ (Oct. 2011).

88. Pidsley, R. *et al.* Critical evaluation of the Illumina MethylationEPIC BeadChip microarray for whole-genome DNA methylation profiling. *Genome Biology* **17,** 208. ISSN: 1474760X. http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1066-1 (Oct. 2016).

89. Steen, H. & Mann, M. *The ABC's (and XYZ's) of peptide sequencing* Sept. 2004. www.nature.com/reviews/molcellbio.

90. Bainbridge, M. N. *et al.* Analysis of the prostate cancer cell line LNCaP transcriptome using a sequencing-by-synthesis approach. *BMC Genomics* **7,** 246. ISSN: 14712164. http://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-7-246 (Sept. 2006).

91. Wang, Z., Gerstein, M. & Snyder, M. *RNA-Seq: A revolutionary tool for transcriptomics* Jan. 2009.

92. Fields, S. & Song, O. K. A novel genetic system to detect protein-protein interactions. *Nature* **340,** 245–246. ISSN: 00280836. https://www.nature.com/articles/340245a0 (1989).

93. De Las Rivas, J. & Fontanillo, C. Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS computational biology* **6,** e1000807. ISSN: 1553-7358. http://www.ncbi.nlm.nih.gov/pubmed/20589078%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2891586 (June 2010).

94. Kamburov, A. *et al.* ConsensusPathDB: Toward a more complete picture of cell biology. *Nucleic Acids Research* **39,** D712–D717. ISSN: 03051048. https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkq1156 (2011).

95. Szklarczyk, D. *et al.* STRING v11: Protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research* **47,** D607–D613. ISSN: 13624962. http://www.ncbi.nlm.nih.gov/pubmed/30476243%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6323986 (Jan. 2019).

96. Mitchell, T. M. *Machine Learning* ISBN: 978-0-07-042807-2 (McGraw-Hill, New York, 1997).

97. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* **60,** 84–90. ISSN: 15577317. http://code.google.com/p/cuda-convnet/ (2017).

98. Zhang, R. & Vadakkepat, P. An Evolutionary Algorithm for Trajectory Based Gait Generation of Biped Robot. *Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems,* 3–8. http://ai.stanford.edu/%7B~%7Drxzhang/An%20Evolutionary%20Algorithm%20for%20Trajectory.pdf (2003).

99. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).

100. Eraslan, G. *et al. Deep learning: new computational modelling techniques for genomics* Apr. 2019. http://www.nature.com/articles/s41576-019-0122-6.

101. Mason, M. J. *et al.* Multiple Myeloma DREAM Challenge reveals epigenetic regulator PHF19 as marker of aggressive disease. *Leukemia* **34,** 1866–1874. ISSN: 14765551 (2020).

102. Salcedo, A. *et al.* A community effort to create standards for evaluating tumor subclonal reconstruction. *Nature Biotechnology* **38,** 97–107. ISSN: 15461696 (2020).

103. Saez-Rodriguez, J. *et al.* Crowdsourcing biomedical research: Leveraging communities as innovation engines. *Nature Reviews Genetics* **17,** 470–486. ISSN: 14710064. http://dx.doi.org/10.1038/nrg.2016.69 (2016).

104. Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* 1st ed. ISBN: 0387310738. http://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738%7B%5C%%7D3FSubscriptionId%7B%5C%%7D3D13CT5CVB80YFWJEPWS02%7B%5C%%7D26tag%7B%5C%%7D3Dws%7B%5C%%7D26linkCode%7B%5C%%7D3Dxm2%7B%5C%%7D26camp%7B%5C%%7D3D2025%7B%5C%%7D26creative%7B%5C%%7D3D165953%7B%5C%%7D26creativeASIN%7B%5C%%7D3D0387310738 (Springer, 2007).

105. Ruder, S. An overview of gradient descent optimization algorithms. arXiv: 1609.04747. http://arxiv.org/abs/1609.04747 (Sept. 2016).

106. Murphy, K. P. *Machine learning: A Probabilistic Perspective* 15. ISBN: 9780262018029 (2012).

107. Hastie, T., Tibshirani, R. & Friedman, J. *The elements of statistical learning: data mining, inference and prediction* 2nd ed. http://www-stat.stanford.edu/%7B~%7Dtibs/ElemStatLearn/ (Springer, 2009).

108. Russel, S. & Norvig, P. *Artificial intelligence—a modern approach 3rd Edition* ISBN: 0136042597. arXiv: 9809069v1 [arXiv:gr-qc] (2012).

109. Freund, Y. & Schapire, R. E. Experiments with a New Boosting Algorithm. *Proceedings of the 13th International Conference on Machine Learning*, 148–156. ISSN: 0706-652X, 1205-7533. http://www.research.att.com/%20http://www.public.asu.edu/%7B~%7Djye02/CLASSES/Fall-2005/PAPERS/boosting-icml.pdf (1996).

110. Ho, T. K. *Random decision forests* in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* **1** (IEEE Computer Society, 1995), 278–282. ISBN: 0818671289.

111. Vapnik, V. N. *An overview of statistical learning theory* 1999.

112. Boser, B. E., Guyon, I. M. & Vapnik, V. N. *Training algorithm for optimal margin classifiers* in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory* (Publ by ACM, 1992), 144–152. ISBN: 089791497X.

113. Schölkopf, B. & Christopher J. C. Burges, Burges, A. J. S. *Advances in Kernel Methods: Support Vector Learning* (eds Scholkopf, B., Burges, C. J. C. & Smola, A. J.) 386 (MIT Press, Cambridge, MA, 1998).

114. Hofmann, T., Schölkopf, B. & Smola, A. J. *Kernel methods in machine learning* 2008. arXiv: 0701907 [math].

115. Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen. *Master's thesis, Institut für Informatik, Technische Universität, Munchen*, 1–71. ISSN: 18168957 18163459. arXiv: 1511.07289. http://people.idsia.ch/%7B~%7Djuergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf (1991).

116. Nair, V. & Hinton, G. E. *Rectified linear units improve Restricted Boltzmann machines* in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning* (2010), 807–814. ISBN: 9781605589077.

117. Glorot, X., Bordes, A. & Bengio, Y. *Deep sparse rectifier neural networks* in *Journal of Machine Learning Research* **15** (2011), 315–323.

118. Lecun, Y., Bengio, Y. & Hinton, G. *Deep learning* May 2015.

119. Sobel, I. & Feldman, G. A 3x3 isotropic gradient operator for image processing. *in Hart, P. E. & Duda R. O. Pattern Classification and Scene Analysis*, 271–272. papers2://publication/uuid/F6C98D8E-0A99-40EF-A91C-0ECA53448D1F (1973).

120. LeCun, Y. *et al.* Gradient Based Learning Applied to Document Recognition. *Proceedings of the IEEE* **86,** 2278–2324. ISSN: 00189219. arXiv: 1102.0183 (1998).

121. Szegedy, C. *et al. Rethinking the Inception Architecture for Computer Vision* in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2016-Decem** (IEEE Computer Society, Dec. 2016), 2818–2826. ISBN: 9781467388504. arXiv: 1512.00567. https://arxiv.org/abs/1512.00567v3.

122. Van Essen, D. C. & Maunsell, J. H. *Hierarchical organization and functional streams in the visual cortex* Jan. 1983.

123. Felleman, D. J. & Van Essen, D. C. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex* **1,** 1–47. ISSN: 14602199 (1991).

124. Aloy, P. & Russell, R. B. *Potential artefacts in protein-interaction networks* Oct. 2002.

125. Rolland, T. *et al.* A proteome-scale map of the human interactome network. *Cell* **159,** 1212–1226. ISSN: 10974172 (Nov. 2014).

126. Hakes, L., Robertson, D. L. & Oliver, S. G. Effect of dataset selection on the topological interpretation of protein interaction networks. *BMC Genomics* **6,** 1–8. ISSN: 14712164 (Sept. 2005).

127. Niepert, M., Ahmed, M. & Kutzkov, K. *Learning Convolutional Neural Networks for Graphs* in *ICML* (2016). ISBN: 9781510829008. arXiv: 1605.05273. http://jmlr.org/proceedings/papers/v48/niepert16.pdf.

128. Defferrard, M., Bresson, X. & Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *NIPS*, 1–14. arXiv: 1606.09375. http://arxiv.org/abs/1606.09375 (2016).

129. Rayward-Smith, V. J. *et al.* Introduction to Algorithms. *The Journal of the Operational Research Society* **42,** 816. ISSN: 01605682. arXiv: 2010(ret.29.4.2010) (1991).

130. Freeman, L. C. A Set of Measures of Centrality Based on Betweenness. *Sociometry* **40,** 35. ISSN: 00380431 (Mar. 1977).

131. Batagelj, V. & Zaveršnik, M. Generalized Cores. *Journal of the ACM* **V,** 1–8. arXiv: 0202039 [cs]. http://arxiv.org/abs/cs/0202039 (Feb. 2002).

132. Henaff, M., Bruna, J. & LeCun, Y. Deep Convolutional Networks on Graph-Structured Data. *arXiv,* 1–10. arXiv: arXiv:1506.05163v1 (2015).

133. Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *ICLR 2017,* 1–10. arXiv: 1609.02907. http://arxiv.org/abs/1609.02907 (2016).

134. Vialatte, J.-C., Gripon, V. & Mercier, G. Generalizing the Convolution Operator to Extend CNNs to Irregular Domains, 1–9. arXiv: 1606.01166. https://arxiv.org/pdf/1606.01166.pdf%20http://arxiv.org/abs/1606.01166 (2016).

135. Jonathan Masci *et al.* ShapeNet: Convolutional Neural Networks on Non-Euclidean Manifolds, 37–45. ISSN: 9780769557205. arXiv: 1501.06297v1. https://infoscience.epfl.ch/record/204949/files/main%7B%5C_%7Dwith%7B%5C_%7Dnames.pdf%20http://arxiv.org/abs/1501.06297v1 (2015).

136. Zhou, J. *et al.* Graph Neural Networks: A Review of Methods and Applications. *arXiv,* 1–20. arXiv: 1812.08434. http://arxiv.org/abs/1812.08434 (Dec. 2018).

137. Shuman, D. I. *et al.* The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* **30,** 83–98. ISSN: 10535888. arXiv: 1211.0053. https://arxiv.org/pdf/1211.0053.pdf (2013).

138. Chung, F. R. *Spectral graph theory. Vol. 92. American Mathematical Soc.* ISBN: 978-0-8218-0315-8 (1997).

139. Bruna, J. *et al.* Spectral Networks and Locally Connected Networks on Graphs. *Iclr,* 14. ISSN: 14644258. arXiv: 1312.6203. http://arxiv.org/abs/1312.6203 (2013).

140. Ortega, A. *et al.* Graph Signal Processing: Overview, Challenges, and Applications. *Proceedings of the IEEE* **106,** 808–828. ISSN: 15582256. arXiv: 1712.00468 (May 2018).

141. Hammond, D. K. & Vandergheynst, P. Wavelets on Graphs via Spectral Graph Theory, 1–37. arXiv: arXiv:0912.3848v1 (2009).

142. Li, Q., Han, Z. & Wu, X.-M. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. *arXiv,* 1–9. arXiv: 1801.07606. http://arxiv.org/abs/1801.07606 (2018).

143. Edwards, M. & Xie, X. Graph Based Convolutional Neural Network, 11. arXiv: 1609.08965. http://www.csvision.swan.ac.uk%20http://arxiv.org/abs/1609.08965 (2016).

144. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529** (2016).

145. Troyanskaya, O. *et al.* Artificial intelligence and cancer. *Nature Cancer* **1,** 149–152. ISSN: 2662-1347. http://www.nature.com/articles/s43018-020-0034-6 (Feb. 2020).

146. Yip, K. Y., Cheng, C. & Gerstein, M. Machine learning and genome annotation: a match meant to be? *Genome biology* **14,** 205. ISSN: 1465-6914. http://download.springer.com/static/pdf/373/art%7B%5C%%7D253A10.1186%7B%5C%%7D252Fgb-2013-14-5-205.pdf?originUrl=http%7B%5C%%7D3A%7B%5C%%7D2F%7B%5C%%7D2Fgenomebiology.biomedcentral.com%7B%5C%%7D2Farticle%7B%5C%%7D2F10.1186%7B%5C%%7D2Fgb-2013-14-5-205%7B%5C&%7Dtoken2=exp=1489507537%7B~%7Dacl=%7B%5C%%7D2Fstatic%7B%5C%%7D2Fpdf%7B%5C%%7D2F373%7B%5C%%7D2Fart%7B%5C%%7D25253A10.1186%7B%5C%%7D25252Fgb (2013).

147. Leung, M. K. K. *et al.* Machine learning in genomic medicine: A review of computational problems and data sets. *Proceedings of the IEEE.* ISSN: 00189219 (2016).

148. Sample, I. *Computer says no: why making AIs fair, accountable and transparent is crucial* 2017. https://www.theguardian.com/science/2017/nov/05/computer-says-no-why-making-ais-fair-accountable-and-transparent-is-crucial.

149. Jeff Larson *et al.* How We Analyzed the COMPAS Recidivism Algorithm - ProPublica. *Propublica,* 1–9. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm (2016).

150. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* **1,** 206–215. ISSN: 2522-5839. http://www.nature.com/articles/s42256-019-0048-x (May 2019).

151. Lapuschkin, S. *et al.* Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications* **10,** 1096. ISSN: 20411723. http://www.nature.com/articles/s41467-019-08987-4 (Dec. 2019).

152. Murdoch, W. J. *et al.* Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America* **116,** 22071–22080. ISSN: 10916490 (Oct. 2019).

153. Hendricks, L. A. *et al. Generating visual explanations* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9908 LNCS** (2016), 3–19. ISBN: 9783319464923. arXiv: 1603.08507.

154. Mittelstadt, B., Russell, C. & Wachter, S. *Explaining explanations in AI* in *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency* (Association for Computing Machinery, Inc, Nov. 2019), 279–288. ISBN: 9781450361255. arXiv: 1811.01439. http://arxiv.org/abs/1811.01439%20http://dx.doi.org/10.1145/3287560.3287574.

155. Cho, H. & Choi, I. InteractionNet: Modeling and Explaining of Noncovalent Protein-Ligand Interactions with Noncovalent Graph Neural Network and Layer-Wise Relevance Propagation. arXiv: 2005.13438. http://arxiv.org/abs/2005.13438 (May 2020).

156. Binder, A. *et al.* Towards computational fluorescence microscopy: Machine learning-based integrated prediction of morphological and molecular tumor profiles. arXiv: 1805.11178. http://arxiv.org/abs/1805.11178 (May 2018).

157. Singh, A., Sengupta, S. & Lakshminarayanan, V. *Explainable deep learning models in medical image analysis* 2020. arXiv: 2005.13799.

158. Bach, S. *et al.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10,** 1–46. ISSN: 19326203. http://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0130140%7B%5C&%7Dtype=printable (2015).

159. Gilpin, L. H. *et al. Explaining explanations: An overview of interpretability of machine learning* in *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018* (Institute of Electrical and Electronics Engineers Inc., May 2019), 80–89. ISBN: 9781538650905. arXiv: 1806.00069. http://arxiv.org/abs/1806.00069.

160. Zeiler, M. D. & Fergus, R. Visualizing and Understanding Convolutional Networks. *Computer Vision–ECCV 2014* **8689,** 818–833. ISSN: 978-3-319-10589-5. arXiv: 1311.2901. https://www.cs.nyu.edu/%7B~%7Dfergus/papers/zeilerECCV2014.pdf%20http://link.springer.com/10.1007/978-3-319-10590-1%7B%5C_%7D53%7B%5C%%7D5Cnhttp://arxiv.org/abs/1311.2901%7B%5C%%7D5Cnpapers3://publication/uuid/44feb4b1-873a-4443-8baa-1730ecd16291 (2014).

161. Olah, C., Mordvintsev, A. & Schubert, L. Feature Visualization. *Distill* **2.** ISSN: 2476-0757. https://distill.pub/2017/feature-visualization (Nov. 2017).

162. Budach, S. & Marsico, A. Pysster: Classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics* **34,** 3035–3037. ISSN: 14602059. http://www.ncbi.nlm.nih.gov/pubmed/29659719%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6129303 (2018).

163. Shrikumar, A., Greenside, P. & Kundaje, A. *Learning important features through propagating activation differences* in *34th International Conference on Machine Learning, ICML 2017* **7** (2017), 4844–4866. ISBN: 9781510855144. arXiv: 1704.02685.

164. Ribeiro, M. T., Singh, S. & Guestrin, C. *"Why should i trust you?" Explaining the predictions of any classifier* in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **13-17-Augu** (2016), 1135–1144. ISBN: 9781450342322. http://dx.doi.org/10.1145/2939672.2939778.

165. Štrumbelj, E. & Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* **41,** 647–665. ISSN: 02193116 (2014).

166. Lapuschkin, S. *Opening the Machine Learning Black Box with Layer-wise Relevance Propagation* PhD thesis (2019), 1–166.

167. Zintgraf, L. M. *et al. Visualizing deep neural network decisions: Prediction difference analysis* in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (2017), 1–12. arXiv: 1702.04595.

168. Baehrens, D. *et al.* How to explain individual classification decisions. *Journal of Machine Learning Research* **11,** 1803–1831. ISSN: 15324435. arXiv: 0912.1128 (2010).

169. Simonyan, K., Vedaldi, A. & Zisserman, A. *Deep inside convolutional networks: Visualising image classification models and saliency maps* in *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings* (2014), 1–8. arXiv: 1312.6034.

170. Sundararajan, M., Taly, A. & Yan, Q. *Axiomatic attribution for deep networks* in *34th International Conference on Machine Learning, ICML 2017* **7** (International Machine Learning Society (IMLS), Mar. 2017), 5109–5118. ISBN: 9781510855144. arXiv: 1703.01365. http://arxiv.org/abs/1703.01365.

171. Ancona, M. *et al. Towards better understanding of gradient-based attribution methods for deep neural networks* in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (International Conference on Learning Representations, ICLR, Nov. 2018). arXiv: 1711.06104. http://arxiv.org/abs/1711.06104.

172. Kindermans, P.-J. *et al.* Investigating the influence of noise and distractors on the interpretation of neural networks. arXiv: 1611.07270. http://arxiv.org/abs/1611.07270 (Nov. 2016).

173. Montavon, G. *et al.* Deep Taylor Decomposition of Neural Networks. *Proceedings of the Workshop on Visualization for Deep Learning at International Conference on Machine Learning (ICML),* 1–3. https://icmlviz.github.io/assets/papers/13.pdf (2016).

174. Montavon, G. *et al.* Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition* **65,** 211–222. ISSN: 00313203. arXiv: 1512.02479 (May 2017).

175. Montavon, G., Samek, W. & Müller, K. R. *Methods for interpreting and understanding deep neural networks* Feb. 2018. arXiv: 1706.07979.

176. Montavon, G. *et al.* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 193–209 (Springer Verlag, 2019).

177. Martin, A. *et al.* TensorFlow: A system for Large-Scale Machine Learning, 265–283. https://ai.google/research/pubs/pub45381 (2016).

178. Shindjalova, R., Prodanova, K. & Svechtarov, V. *Modeling data for tilted implants in grafted with bio-oss maxillary sinuses using logistic regression* in *AIP Conference Proceedings* **1631** (Dec. 2014), 58–62. ISBN: 9780735412705. arXiv: 1412.6980. http://arxiv.org/abs/1412.6980.

179. Leek, J. T. *et al. Tackling the widespread and critical impact of batch effects in high-throughput data* Oct. 2010. /pmc/articles/PMC3880143/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3880143/.

180. Michels, K. B. *et al.* Recommendations for the design and analysis of epigenome-wide association studies. *Nature Methods* **10,** 949–955. ISSN: 15487091 (2013).

181. Cibulskis, K. *et al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology* **31,** 213–219. ISSN: 10870156 (2013).

182. Benjamin, D. *et al.* Calling Somatic SNVs and Indels with Mutect2. https://doi.org/10.1101/861054 (2019).

183. Fan, Y. *et al.* MuSE: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data. *Genome biology* **17,** 178. ISSN: 1474760X. http://dx.doi.org/10.1186/s13059-016-1029-6 (2016).

184. Shah, S. P. *et al.* Mutational evolution in a lobular breast tumour profiled at single nucleotide resolution. *Nature* **461,** 809–813. ISSN: 00280836 (2009).

185. Hao, D., Wang, L. & Di, L. J. Distinct mutation accumulation rates among tissues determine the variation in cancer risk. *Scientific Reports* **6,** 1–5. ISSN: 20452322 (Jan. 2016).

186. Frankish, A. *et al.* GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Research* **47,** D766–D773. ISSN: 0305-1048. http://www.ncbi.nlm.nih.gov/pubmed/30357393%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6323946%20https://academic.oup.com/nar/article/47/D1/D766/5144133 (Jan. 2019).

187. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8,** 118–127. ISSN: 14654644. https://academic.oup.com/biostatistics/article/8/1/118/252073 (Jan. 2007).

188. Razick, S., Magklaras, G. & Donaldson, I. M. iRefIndex: A consolidated protein interaction database with provenance. *BMC Bioinformatics* **9,** 405. ISSN: 14712105. http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-405 (Sept. 2008).

189. Khurana, E. *et al.* Interpretation of Genomic Variants Using a Unified Biological Network Approach. *PLoS Computational Biology* **9** (ed Rzhetsky, A.) e1002886. ISSN: 1553734X. http://dx.plos.org/10.1371/journal.pcbi.1002886 (Mar. 2013).

190. Huang, J. K. *et al.* Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. *Cell Systems* **6,** 484–495.e5. ISSN: 24054720 (2018).

191. Liu, S. H. *et al.* DriverDBv3: A multi-omics database for cancer driver gene research. *Nucleic Acids Research* **48,** D863–D870. ISSN: 13624962 (Jan. 2020).

192. Liu, Y., Sun, J. & Zhao, M. *ONGene: A literature-based database for human oncogenes* Feb. 2017.

193. Chakravarty, D. *et al.* OncoKB: A Precision Oncology Knowledge Base. *JCO Precision Oncology* **1,** 1–16. ISSN: 2473-4284 (Nov. 2017).

194. Repana, D. *et al.* The Network of Cancer Genes (NCG): a comprehensive catalogue of known and candidate cancer genes from cancer sequencing screens. *Genome Biology* **20,** 1–12. ISSN: 1474-760X. https://genomebiology.biomedcentral.com/articles/10.1186/s13059-018-1612-0 (2019).

195. Sondka, Z. *et al.* The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers. *Nature Reviews Cancer* **18,** 696–705. ISSN: 14741768. http://www.nature.com/articles/s41568-018-0060-1 (2018).

196. Futreal, P. A. *et al. A census of human cancer genes* 2004.

197. Zhao, M., Sun, J. & Zhao, Z. TSGene: A web resource for tumor suppressor genes. *Nucleic Acids Research* **41.** ISSN: 03051048 (Jan. 2013).

198. Bailey, M. H. *et al.* Comprehensive Characterization of Cancer Driver Genes and Mutations. *Cell* **173,** 371–385.e18. ISSN: 1097-4172. http://www.ncbi.nlm.nih.gov/pubmed/29625053%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6029450 (Apr. 2018).

199. Kim, J. *et al.* DigSee: Disease gene search engine with evidence sentences (version cancer). *Nucleic acids research* **41.** ISSN: 13624962. http://gcancer.org/digsee. (2013).

200. McKusick, V. A. Mendelian Inheritance in Man and its online version, OMIM. *American journal of human genetics* **80,** 588–604. ISSN: 0002-9297. http://www.ncbi.nlm.nih.gov/pubmed/17357067%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1852721 (Apr. 2007).

201. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research* **28,** 27–30. ISSN: 0305-1048. http://www.ncbi.nlm.nih.gov/pubmed/10592173%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC102409 (Jan. 2000).

202. Liberzon, A. *et al.* The Molecular Signatures Database Hallmark Gene Set Collection. *Cell Systems* **1,** 417–425. ISSN: 2405-4712. https://www.sciencedirect.com/science/article/pii/S2405471215002185 (Dec. 2015).

203. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* **102,** 15545–15550. ISSN: 00278424 (Oct. 2005).

204. MacKay, D. *Information Theory, Inference, and Learning Algorithms* 2004.

205. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58,** 267–288. ISSN: 00359246. http://www.jstor.org/stable/2346178 (1996).

206. Hoerl, A. E. & Kennard, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **12,** 55–67. ISSN: 0040-1706. http://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634 (Feb. 1970).

207. Srivastava, N. *et al.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15,** 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html (2014).

208. Rojas, R. *Neural Networks: A Systematic Introduction* ISBN: 3540605053 (Springer-Verlag, Berlin, Heidelberg, 1996).

209. Sixt, L., Granz, M. & Landgraf, T. *When explanations lie: Why many modified BP attributions fail* 2019. arXiv: 1912.09818. https://www.researchgate.net/publication/338115768.

210. Schwarzenberg, R. *et al. Layerwise relevance visualization in convolutional text graph classifiers* in *EMNLP-IJCNLP 2019 - Graph-Based Methods for Natural Language Processing - Proceedings of the 13th Workshop* (Sept. 2019), 58–62. ISBN: 9781950737864. arXiv: 1909.10911. http://arxiv.org/abs/1909.10911.

211. Hu, J., Li, T. & Dong, S. GCN-LRP explanation: exploring latent attention of graph convolutional networks, 1–8 (2020).

212. XIE, S. & LU, M. *Interpreting and understanding graph convolutional neural network using gradient-based attribution method* 2019. arXiv: 1903.03768.

213. Jeong, H. *et al.* Lethality and centrality in protein networks. *Nature* **411,** 41–42. ISSN: 00280836. arXiv: 0105306 [cond-mat] (May 2001).

214. Krogan, N. J. *et al.* Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature* **440,** 637–643. ISSN: 00280836 (2006).

215. Liu, G., Wong, L. & Chua, H. N. Complex discovery from weighted PPI networks. *Bioinformatics* **25,** 1891–1897. ISSN: 13674803. https://academic.oup.com/bioinformatics/article/25/15/1891/211634 (Aug. 2009).

216. Adamcsek, B. *et al.* CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics* **22,** 1021–1023. ISSN: 13674803. http://angel.elte.hu/clustering. (Apr. 2006).

217. Schaefer, M. H., Serrano, L. & Andrade-Navarro, M. A. 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.** ISSN: 16648021 (2015).

218. Mrowka, R., Patzak, A. & Herzel, H. Is there a bias in proteome research? *Genome Research* **11,** 1971–1973. ISSN: 10889051 (Dec. 2001).

219. Fabian, P. *et al. Scikit-learn: Machine Learning in Python* tech. rep. (2011), 2825–2830. http://scikit-learn.sourceforge.net..

220. Mikolov, T. *et al. Efficient estimation of word representations in vector space* in *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings* (International Conference on Learning Representations, ICLR, Jan. 2013). arXiv: 1301.3781. https://arxiv.org/abs/1301.3781v3.

221. Page, L. *et al.* The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems,* 1–17. ISSN: 1752-0509. arXiv: 1111.4503v1. http://storm.cis.fordham.edu/%7B~%7Dgweiss/selected-papers/classic-pagerank-paper.pdf%20http://ilpubs.stanford.edu:8090/422 (1998).

222. Perozzi, B., Al-Rfou, R. & Skiena, S. DeepWalk: Online Learning of Social Representations. *arXiv,* 1–10. arXiv: 1403.6652. http://arxiv.org/abs/1403.6652%20http://dx.doi.org/10.1145/2623330.2623732 (Mar. 2014).

223. Fodde, R. The APC gene in colorectal cancer. *European Journal of Cancer* **38,** 867–871. ISSN: 09598049. http://www.ncbi.nlm.nih.gov/pubmed/11978510 (May 2002).

224. Zhao, Z. *et al.* Multiple biological functions of Twist1 in various cancers. *Oncotarget* **8,** 20380–20393. ISSN: 1949-2553. http://www.ncbi.nlm.nih.gov/pubmed/28099910%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5386770 (Mar. 2017).

225. Khan, M. A. *et al. Twist: A molecular target in cancer therapeutics* Oct. 2013. http://www.ncbi.nlm.nih.gov/pubmed/23873099.

226. Gajula, R. P. *et al.* Structure-Function Studies of the bHLH Phosphorylation Domain of TWIST1 in Prostate Cancer Cells. *Neoplasia* **17,** 16–31. ISSN: 14765586. https://pubmed.ncbi.nlm.nih.gov/25622896/%20https://pubmed.ncbi.nlm.nih.gov/25622896/?dopt=Abstract (Jan. 2015).

227. Malek, R. *et al.* TWIST1-WDR5-hottip regulates Hoxa9 chromatin to facilitate prostate cancer metastasis. *Cancer Research* **77,** 3181–3193. ISSN: 15387445. http://cancerres.aacrjournals.org/ (June 2017).

228. Patwardhan, D. *et al. STIL balancing primary microcephaly and cancer* Jan. 2018.

229. Schuijers, J. *et al.* Transcriptional Dysregulation of MYC Reveals Common Enhancer-Docking Mechanism. *Cell Reports* **23,** 349–360. ISSN: 22111247. http://www.ncbi.nlm.nih.gov/pubmed/29641996%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5929158 (Apr. 2018).

230. Lopes-Ramos, C. M. *et al.* E2F1 somatic mutation within miRNA target site impairs gene regulation in colorectal cancer. *PLoS ONE* **12.** ISSN: 19326203 (July 2017).

231. Jinesh, G. G. *et al. Molecular genetics and cellular events of K-Ras-driven tumorigenesis* Feb. 2018. http://www.ncbi.nlm.nih.gov/pubmed/29059163%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5817384.

232. Goodman, R. H. & Smolik, S. *CBP/p300 in cell growth, transformation, and development* July 2000.

233. Shen, W.-f. *et al.* The HOX Homeodomain Proteins Block CBP Histone Acetyltransferase Activity. *Molecular and Cellular Biology* **21,** 7509–7522. ISSN: 0270-7306 (Nov. 2001).

234. Svingen, T. & Tonissen, K. F. *Hox transcription factors and their elusive mammalian gene targets* Aug. 2006.

235. Eisfeld, A. K. *et al.* NRAS isoforms differentially affect downstream pathways, cell growth, and cell transformation. *Proceedings of the National Academy of Sciences of the United States of America* **111,** 4179–4184. ISSN: 10916490. https://pubmed.ncbi.nlm.nih.gov/24586049/ (Mar. 2014).

236. Irahara, N. *et al.* NRAS mutations are rare in colorectal cancer. *Diagnostic Molecular Pathology* **19,** 157–163. ISSN: 10529551. /pmc/articles/PMC2929976/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2929976/ (Sept. 2010).

237. Hobbs, G. A., Der, C. J. & Rossman, K. L. RAS isoforms and mutations in cancer at a glance. *Journal of Cell Science* **129,** 1287–1292. ISSN: 14779137. https://pubmed.ncbi.nlm.nih.gov/26985062/ (Apr. 2016).

238. Nevins, J. R. The Rb/E2F pathway and cancer. *Human Molecular Genetics* **10,** 699–703. ISSN: 0964-6906. http://www.ncbi.nlm.nih.gov/pubmed/11257102 (Apr. 2001).

239. Li, Y. & Seto, E. HDACs and HDAC inhibitors in cancer development and therapy. *Cold Spring Harbor Perspectives in Medicine* **6.** ISSN: 21571422. http://www.ncbi.nlm.nih.gov/pubmed/27599530%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5046688 (Oct. 2016).

240. Luo, R. X., Postigo, A. A. & Dean, D. C. Rb interacts with histone deacetylase to repress transcription. *Cell* **92,** 463–473. ISSN: 00928674 (Feb. 1998).

241. Mittal, P. & Roberts, C. W. The SWI/SNF complex in cancer — biology, biomarkers and therapy. *Nature Reviews Clinical Oncology* **17,** 435–448. ISSN: 17594782. http://dx.doi.org/10.1038/s41571-020-0357-3 (2020).

242. Tsurusaki, Y. *et al.* Mutations affecting components of the SWI/SNF complex cause Coffin-Siris syndrome. *Nature Genetics* **44,** 376–378. ISSN: 10614036 (2012).

243. Sausen, M. *et al.* Integrated genomic analyses identify ARID1A and ARID1B alterations in the childhood cancer neuroblastoma. *Nature Genetics* **45,** 12–17. ISSN: 10614036. https://pubmed.ncbi.nlm.nih.gov/23202128/ (Jan. 2013).

244. Xu, F. *et al. Roles of the PI3K/AKT/mTOR signalling pathways in neurodegenerative diseases and tumours* Apr. 2020. https://doi.org/10.1186/s13578-020-00416-0.

245. Frieze, A., Horn, P. & Pralat, P. *Algorithms and Models for the Web-Graph: 8th International Workshop, WAW 2011, Atlanta, GA, USA, May 27-29, 2011, Proceedings* ISBN: 9783642212864. https://books.google.de/books?id=HoKrCAAAQBAJ (Springer Berlin Heidelberg, 2011).

246. Miller, J. C. & Hagberg, A. *Efficient generation of networks with given expected degrees* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **6732 LNCS** (Springer, Berlin, Heidelberg, 2011), 115–126. ISBN: 9783642212857. https://link.springer.com/chapter/10.1007/978-3-642-21286-4%7B%5C_%7D10.

247. Holme, P. & Kim, B. J. Growing scale-free networks with tunable clustering. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **65,** 026107. ISSN: 1063651X. arXiv: 0110452 [cond-mat]. http://www.ncbi.nlm.nih.gov/pubmed/11863587 (Feb. 2002).

248. Lytle, N. K., Barber, A. G. & Reya, T. Stem cell fate in cancer growth, progression and therapy resistance. *Nature Reviews Cancer* **18,** 669–680. ISSN: 14741768. http://dx.doi.org/10.1038/s41568-018-0056-x (2018).

249. Tsherniak, A. *et al.* Defining a Cancer Dependency Map. *Cell* **170,** 564–576.e16. ISSN: 10974172. http://dx.doi.org/10.1016/j.cell.2017.06.010http://dx.doi.org/10.1016/j.cell.2017.06.010 (2017).

250. Meyers, R. M. *et al.* Computational correction of copy number effect improves specificity of CRISPR-Cas9 essentiality screens in cancer cells. *Nature Genetics* **49,** 1779–1784. ISSN: 15461718 (2017).

251. Eisenberg, E. & Levanon, E. Y. *Human housekeeping genes, revisited* Oct. 2013. https://pubmed.ncbi.nlm.nih.gov/23810203/.

252. Kluger, Y. *et al. Spectral biclustering of microarray data: Coclustering genes and conditions* Apr. 2003.

253. Von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing* **17,** 395–416. ISSN: 09603174. arXiv: 0711.0189. www.springer.com. (2007).

254. Wee, Y. *et al.* Identification of novel prognosis-related genes associated with cancer using integrative network analysis. *Scientific Reports* **8.** ISSN: 20452322 (Dec. 2018).

255. Priestley, P. *et al.* Pan-cancer whole-genome analyses of metastatic solid tumours. *Nature* **575,** 210–216. ISSN: 14764687 (Nov. 2019).

256. Liu, Y. *et al.* CIGene: A literature-based online resource for cancer initiation genes. *BMC Genomics* **19.** ISSN: 14712164 (July 2018).

257. Suvà, M. L., Riggi, N. & Bernstein, B. E. *Epigenetic reprogramming in cancer* Mar. 2013.

258. Keita, M. *et al.* Global methylation profiling in serous ovarian cancer is indicative for distinct aberrant DNA methylation signatures associated with tumor aggressiveness and disease progression. *Gynecologic Oncology* **128,** 356–363. ISSN: 00908258. http://dx.doi.org/10.1016/j.ygyno.2012.11.036 (2013).

259. Webber, B. R. *et al.* DNA methylation of Runx1 regulatory regions correlates with transition from primitive to definitive hematopoietic potential in vitro and in vivo. *Blood* **122,** 2978–2986. ISSN: 15280020. https://pubmed.ncbi.nlm.nih.gov/24030384/ (2013).

260. Tarjan, R. *Depth- first search and linear graph algorithms* in (1971), 114–121.

261. Bissell, M. J. & Hines, W. C. *Why don't we get more cancer? A proposed role of the microenvironment in restraining cancer progression* Mar. 2011. https://pubmed.ncbi.nlm.nih.gov/21383745/.

262. Yu, Y. *et al.* The inhibitory effects of COL1A2 on colorectal cancer cell proliferation, migration, and invasion. *Journal of Cancer* **9,** 2953–2962. ISSN: 18379664. https://pubmed.ncbi.nlm.nih.gov/30123364/ (2018).

263. Sigismund, S., Avanzato, D. & Lanzetti, L. *Emerging functions of the EGFR in cancer* Jan. 2018. https://pubmed.ncbi.nlm.nih.gov/29124875/.

264. Veeriah, S. *et al.* The tyrosine phosphatase PTPRD is a tumor suppressor that is frequently inactivated and mutated in glioblastoma and other human cancers. *Proceedings of the National Academy of Sciences of the United States of America* **106,** 9435–9440. ISSN: 00278424. www.pnas.org/cgi/content/full/ (June 2009).

265. Oh, E. S. *et al. Cell adhesion in cancer* 2012. https://pubmed.ncbi.nlm.nih.gov/22536252/.

266. Xing, P. *et al. Roles of low-density lipoprotein receptor-related protein 1 in tumors* Jan. 2016. https://pubmed.ncbi.nlm.nih.gov/26738504/.

267. Pu, X. *et al.* Caspase-3 and caspase-8 expression in breast cancer: caspase-3 is associated with survival. *Apoptosis* **22,** 357–368. ISSN: 1573675X. https://pubmed.ncbi.nlm.nih.gov/27798717/ (Mar. 2017).

268. Schramek, D. *et al.* Direct in vivo RNAi screen unveils myosin IIa as a tumor suppressor of squamous cell carcinomas. *Science* **343,** 309–313. ISSN: 10959203. https://science.sciencemag.org/content/343/6168/309%20https://science.sciencemag.org/content/343/6168/309.abstract (Jan. 2014).

269. Wang, B. *et al.* MYH9 promotes growth and metastasis via activation of MAPK/AKT signaling in colorectal cancer. *Journal of Cancer* **10,** 874–884. ISSN: 18379664. https://pubmed.ncbi.nlm.nih.gov/30854093/ (2019).

270. Chen, R. *et al. Histone methyltransferase SETD2: A potential tumor suppressor in solid cancers* 2020. /pmc/articles/PMC7097956/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7097956/.

271. Klink, B. U. *et al.* Structure of the human BBSome core complex. *eLife* **9.** ISSN: 2050084X (Jan. 2020).

272. Yang, K. *et al.* Integrative analysis reveals CRHBP inhibits renal cell carcinoma progression by regulating inflammation and apoptosis. *Cancer Gene Therapy* **27,** 607–618. ISSN: 14765500 (Aug. 2020).

273. Deng, L. *et al. The role of ubiquitination in tumorigenesis and targeted drug discovery* Dec. 2020. https://pubmed.ncbi.nlm.nih.gov/32296023/.

274. Sánchez-Martın, P. & Komatsu, M. *p62/SQSTM1 – Steering the cell through health and disease* Nov. 2018. https://pubmed.ncbi.nlm.nih.gov/30397181/.

275. Li, Y. *et al.* LRP6 expression promotes cancer cell proliferation and tumorigenesis by altering β-catenin subcellular distribution. *Oncogene* **23,** 9129–9135. ISSN: 09509232 (2004).

276. Ding, Y. *et al.* Caprin-2 enhances canonical Wnt signaling through regulating LRP5/6 phosphorylation. *Journal of Cell Biology* **182,** 865–872. ISSN: 00219525 (Sept. 2008).

277. Tombran-Tink, J. & Barnstable, C. J. PEDF: A multifaceted neurotrophic factor. *Nature Reviews Neuroscience* **4,** 628–636. ISSN: 14710048. www.nature.com/reviews/neuro (2003).

278. Ivanov, A. A., Khuri, F. R. & Fu, H. *Targeting protein-protein interactions as an anticancer strategy* 2013.

279. Mourikis, T. P. *et al.* Patient-specific cancer genes contribute to recurrently perturbed pathways and establish therapeutic vulnerabilities in esophageal adenocarcinoma. *Nature Communications* **10,** 3101. ISSN: 2041-1723. http://www.nature.com/articles/s41467-019-10898-3 (Dec. 2019).

280. Lai, A. Y. & Wade, P. A. *Cancer biology and NuRD: A multifaceted chromatin remodelling complex* Aug. 2011.

281. Shi, J. *et al.* YWHAZ promotes ovarian cancer metastasis by modulating glycolysis. *Oncology Reports* **41,** 1101–1112. ISSN: 17912431. http://www.ncbi.nlm.nih.gov/pubmed/30535456 (Feb. 2019).

282. Giannakis, M. *et al.* Genomic Correlates of Immune-Cell Infiltrates in Colorectal Carcinoma. *Cell Reports* **15,** 857–865. ISSN: 22111247. http://www.ncbi.nlm.nih.gov/pubmed/27149842%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4850357 (Apr. 2016).

283. McInnes, L., Healy, J. & Melville, J. *UMAP: Uniform manifold approximation and projection for dimension reduction* Feb. 2018. arXiv: 1802.03426. http://arxiv.org/abs/1802.03426.

284. Gerstberger, S., Hafner, M. & Tuschl, T. A census of human RNA-binding proteins. *Nature Reviews Genetics* **15,** 829–845. ISSN: 14710064 (2014).

285. Mitchell, S. F. & Parker, R. *Principles and Properties of Eukaryotic mRNPs* May 2014.

286. Hentze, M. W. *et al.* A brave new world of RNA-binding proteins. *Nature Reviews Molecular Cell Biology* **19,** 327–341. ISSN: 1471-0072. http://www.nature.com/doifinder/10.1038/nrm.2017.130 (Jan. 2018).

287. LESLIE, C., ESKIN, E. & NOBLE, W. S. *THE SPECTRUM KERNEL: A STRING KERNEL FOR SVM PROTEIN CLASSIFICATION* in *Biocomputing 2002* (2001), 564–575. ISBN: 978-981-02-4777-5. http://www.ncbi.nlm.nih.gov/pubmed/11928508%20http://www.worldscientific.com/doi/abs/10.1142/9789812799623%7B%5C_%7D0053.

288. Calabretta, S. & Richard, S. *Emerging Roles of Disordered Sequences in RNA-Binding Proteins* 2015. https://pubmed.ncbi.nlm.nih.gov/26481498/.

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AA	Amino Acid
ANN	Artificial Neural Network
AUPRC	Area under the Precision-Recall Curve
bp	Base Pair
CCG	Candidate Cancer Gene
cDNA	complementary DNA
CNA	Copy Number Aberration
CNN	Convolutional Neural Network
CPDB	Consensus Path DB
CpG	Cytosine followed by Guanine
CRISPRi	CRISPR interference
CV	Cross-Validation
DNA	Deoxyribonucleic acid
DNN	Deep Neural Network
FPKM	Fragments Per Kilobase Million
GCN	Graph Convolutional Network
GD	Gradient Descent
GFT	graph Fourier transform
GOEA	Gene Ontology Enrichment Analysis
GPU	Graphics Processing Unit
GSP	Graph Signal Processing
GTEx	Genotype-Tissue Expression
HP	Hyper-Parameter
HTS	High Throughput Sequencing
ICGC	International Cancer Genome Consortium

KCG	Known Cancer Gene
LRP	Layer-wise relevance propagation
ML	Machine Learning
mRNA	messenger RNA
MSE	mean-squared error
NCG	Network of Cancer Genes
NGS	Next-Generation Sequencing
NPCG	Newly Predicted Cancer Gene
PCAWG	Pan-Cancer Analysis of Whole Genomes
PPI	Protein-Protein-Interaction
ReLU	rectified linear unit
RNA	Ribonucleic acid
RNA-seq	RNA sequencing
RNAi	RNA interference
RWR	Random Walk with Restart
SCC	Strongly-Connected Component
SNV	Single Nucleotide Variant
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TAP-MS	Tandem Purification coupled to Mass Spectrometry
TCGA	The Cancer Genome Atlas
TF	Transcription Factor
TFBS	Transcription Factor Binding Site
TPU	Tensor Processing Unit
TSG	Tumor Suppressor Gene
TSS	Transcription Start Site
WES	Whole Exome Sequencing
Y2H	Yeast Two-Hybrid

Education

2017 – 2020, Max Planck Institute for molecular genetics
PhD student in the research group of Annalisa Marsico. *Title of the Thesis: "Integration of multi-omics data with graph convolutional networks to identify cancer-associated genes".*
2013 – 2016, Freie Universität Berlin
Master of Science in Computer Science with focus on artificial intelligence, robotics and machine learning. Main Subjects: Machine Learning, Pattern recognition, image processing. *Title of the Thesis: "Detecting structural specifics of DHSs from sequences alone using convolutional Deep Belief Networks".*
2012 - 2013, Université Paris Diderot - Paris 7
Study abroad with focus on artificial intelligence and machine learning.
2009 - 2013, Freie Universität Berlin
Bachelor of Science in Computer Science.
Bachelor thesis on self localization of humanoid autonomous soccer playing robots.
2005 - 2006, Western Springs College (New Zealand)
Study abroad program in Auckland, NZ.
2001 - 2008, Marie-Curie-Gymnasium Berlin
Secondary School.

Experience

Jan 2016 - Oct 2016, Max Planck Institute for molecular genetics
Student assistant.
Developing GPU based solutions for deep learning applications using Theano and Python.
Oct 2015 - Dec 2015, Humboldt Universität zu Berlin
Student assistant.
Working focus was on tracking movements in ultrasound videos using MATLAB.
2013 - 2015, FUmanoids, the soccer playing robots
Honorary work in the student's initiative on soccer playing robots.
Work involved image recognition and classification with SVMs, development of a localization framework and development of a mathematical model for strategy based decision making.
2015, Hefei, China
Participation in the RoboCup competition.
`www.robocup2015.org`
2014, Teheran, Iran
Participation in the IranOpen competition.
2009, IVU Traffic Technologies AG
Internship in a company focusing on software solutions for public transport.
2008 - 2009, New Zealand

One year abroad.

Languages

German Mother tongue
English Fluent
French Fluent

References

References upon request.

SUMMARY

Cancer is thought to arise from the accumulation of genetic changes in the DNA of the patient. Mutations can occur during replication of cells or from external factors. Given the current knowledge of gene regulation it is not yet possible to link cancer phenotypes directly to the genetic alterations.
Despite the vast increase of available high-throughput molecular data, the *in silico* identification of disease genes for multi-factorial diseases such as cancer is still a challenging task. Perturbation of entire modules in cellular networks, and genetic, as well as non-genetic gene alternations, contribute to tumorigenesis. This necessitates the development of predictive models able to effectively integrate and process different data modalities. Most approaches cannot combine multi-dimensional molecular data with gene-gene interactions and the few methods that achieve that are hard to interpret.
In this thesis, I introduce EMOGI, an explainable machine learning method based on Graph Convolutional Networks (GCNs) to predict cancer genes by combining multi-omics data, such as mutations, copy number changes, DNA methylation and gene expression profiles across different cancers, together with Protein-Protein Interaction (PPI) networks. By profiting from different data representations, EMOGI was more accurate than previous methods in predicting known cancer genes, with an average increase in area under the precision-recall curve of 3% – 37% across different PPI networks and data sets.
We applied the Layer-Wise Relevance Propagation (LRP) technique to learn the molecular features that contributed to the classification of each individual cancer gene. We also identified relevant cancer modules in the PPI network, and stratified genes according to whether their classification was mainly driven by the interactome, mutation rate or alterations in either DNA methylation or gene expression. We propose a new high-confidence list of 165 putative novel cancer genes which do not harbour recurrent alterations, but rather participate in PPIs with well-known cancer drivers. We functionally validated those novel predictions with publicly available loss-of-function screens. We believe that our results might open new diagnostic and therapeutic avenues in precision oncology, and that our method can applied to predict biomarkers for other complex diseases.

ZUSAMMENFASSUNG

Krebserkrankungen sind die Folge von genetischen Veränderungen in der DNA des Patienten. Diese Mutationen entstehen durch Fehler bei der Replikation oder äußeren Einflüssen und nehmen im Laufe des Lebens zu. Mit dem bisherigen Kenntnisstand der Genregulation ist es jedoch nicht möglich, alle Mutationen direkt mit dem Phenotyp in Verbindung zu bringen.
Die stetig wachsende Masse an sequenz-basierten Daten von Tumoren und normalen Zellen hat die Herausforderungen an computergestützte Methoden enorm verändert. Insbesondere die Identifikation von Genen, die mit Krankheiten assoziiert werden können, ist nach wie vor komplex. Insbesondere bei Krebserkrankungen und anderen komplexen Krankheiten sind es meist Kombinationen von vielen verschiedenen Mutationen, epigenetischen Veränderungen und die Perturbation von Modulen in zellulären Netzwerken, die zur Erkrankung führen oder für Resistenzen verantwortlich sind.
Daher ist es zwingend notwendig, statistische Methoden zur Priorisierung und Vorhersage von krebs-assoziierten Genen zu entwickeln, welche verschiedene Typen von Daten integrieren, um die komplexen Dynamiken in Krebserkrankungen zu berücksichtigen. Die meisten bisher entwickelten Methoden fokussieren sich entweder nur auf genetische Veränderungen im Tumor (sog. Copy Number Alterations oder einzelne Mutationen von Basenpaaren), vernachlässigen zelluläre Netzwerke von interagierenden Proteinen oder sind nicht interpretierbar.
In dieser Arbeit präsentiere ich EMOGI, eine interpretierbare Methode des maschinellen Lernens, welche auf Graph Convolutional Networks (GCNs) basiert. GCNs stellen eine Erweiterung der Konvolutions-Netze da, welche Graph-basierte Daten mit hochdimensionalen Attributen der einzelnen Knoten verknüpfen und in ihre Vorhersagen einbeziehen. Ich nutze EMOGI, um krebs-assoziierte Gene zu identifizieren, die im Nachgang in gezielten Hypothesen-basierten Studien verifiziert werden können. EMOGI integriert genetische Daten (Punktmutationen und strukturelle Variationen), DNA Methylierung und Transkriptions-Daten für 16 verschiedene Krebsarten mit Protein-Protein Interaktionsnetzwerken und ist akkurater als bisherige Methoden.
Außerdem nutze ich die Interpretations-Methode Layer-wise Relevance Propagation (LRP) für neuronale Netze, um die Klassifizierung einzelner Gene *a posteriori* zu verstehen. Mithilfe von LRP können diejenigen Attribute von Genen und auch Protein-Protein Interaktionen identifiziert werden, die für die Klassifikation eines Gens verantwortlich waren. Anschließend können wir die LRP Erklärungen für einzelne Gene in einer Cluster-Analyse gruppieren und finden Gruppen von Genen, die ähnlichen Prinzipien im Tumor unterliegen. Wir schlagen 165 Gene vor, die bisher nicht mit Krebs in Verbindung gebracht worden sind und zeigen, dass diese mit bekannten Krebsgenen interagieren und interessanterweise außerdem für das Überleben von Krebszelllinien notwendig sind.
Ich glaube, dass EMOGI eine wertvolle Methode ist und die erzielten Ergebnisse neue Möglichkeiten in der Diagnose und Therapie von Krebserkrankungen darstellen kön-

nen. Die hier vorgestellte Methode ist nicht auf die Anwendung von Krebserkrankungen beschränkt, sondern kann für andere komplexe Krankheiten, für die ähnlich viele Datenmengen vorhanden sind, genutzt werden.