SAPIENZA
UNIVERSITÀ DI ROMA

Toward Explainable Biomedical Deep Learning

Training and Explaining Neural Networks in Bioinformatics and Medicinal Chemistry

Faculty of Information Engineering, Informatics and Statistics
Department of Computer, Control and Management Engineering
Ph.D. in Data Science (XXXV cycle)

Andrea Mastropietro
ID number 1652886

Advisor
Prof. Aris Anagnostopoulos

Co-Advisor
Dr. Paolo Tieri

Academic Year 2022/2023

Thesis defended on January 31st, 2024

Toward Explainable Biomedical Deep Learning
PhD thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: mastropietro@diag.uniroma1.it

Research is the purest expression of inner freedom

To Leo

Abstract

Deep learning has been extensively utilized in the domains of bioinformatics and chemoinformatics, yielding compelling results. However, neural networks have predominantly been regarded as black boxes, characterized by internal mechanisms that hinder interpretability due to the highly nonlinear functions they learn. In the biomedical field, this lack of interpretability is undesirable, as it is imperative for scientists to comprehend the reasons behind the occurrence of specific diseases or the molecular properties that make a compound effective against a particular target protein. Consequently, the inherent closure of those models keeps their results far from being trusted. To address this issue and make deep learning suitable for bioinformatics and chemoinformatics tasks, there is the urge to develop techniques for explainable artificial intelligence (XAI). These techniques should be capable of measuring the significance of input features for predictions or determining the strength of their interactions. The ability to provide explanations must be integrated into the biomedical deep learning pipeline, which utilizes available data sources to uncover new insights regarding potentially disease-associated genes, thereby facilitating the repurposing and development of new drugs. In line with this objective, this thesis focuses on the development of innovative explainability techniques for neural networks and demonstrates their effective applications in bioinformatics and medicinal chemistry. The devised models find their place in the pipeline, wherein each component of the protocol generates effective and explainable results. These results span from the discovery of disease genes to the repurposing and development of drugs. However, deep learning lives in synergy with classical machine learning models and network-based algorithms, which remain relevant in this field and, therefore, hold a place within this thesis. Moreover, they offer the basis for proper training of deep learning models and pave the way for the development of XAI techniques for neural networks. The proposed work demonstrates how XAI can benefit biomedicine, proving deep learning to be a powerful tool to solve biomedical problems and that the obtained results can be explained. This contributes to the delivery of not only accurate but also trustworthy results, fulfilling the need for explainability of medical doctors, geneticists, and scientists in the life sciences and leading toward a fully explainable biomedical deep learning pipeline.

Contents

Chapter 1

Introduction

In recent years, deep learning [1] has raised the interest of scientists and practitioners in the broader field of biomedicine. Given its performances, often superior to standard machine learning models and statistical techniques, researchers focused deeply on developing and applying neural networks in bioinformatics, chemoinformatics, and medicinal chemistry. However, neural networks have almost always been treated as black boxes, as high-performing machine learning tools that can deliver accurate results with no clear idea of how to interpret them. In fact, the highly nonlinear functions learned by those models prevent the possibility of opening the model itself and providing an interpretation of the results, unlike simpler models such as linear regression and decision trees, from which one can extract the rules determining the final prediction.

This behavior is not desirable in medical scenarios, in which medical doctors and researchers are not only interested in the occurrence of a phenomenon (e.g., the likelihood of a disease) but they need to know why the phenomenon happens, what input features drove the prediction (e.g., the presence of a mutated gene). The closure of those models kept the results obtained far from being trusted and limited their effective use in medical fields [2, 3], such as genetics [4]. In this regard, feature importance and correlation between input and output become essential when using machine learning models in life sciences. Given the huge interest in using deep learning in this context, there has been a rising need for the development of explainable artificial intelligence (XAI) [5] techniques that could cope with the absence of interpretability. Even though neural networks are the most prominent example of black-box machine learning models, this lack of interpretability is by no means confined to deep learning but also invests other "classical" machine learning algorithms, like support vector machines (SVMs) [6, 7]. Although support vectors can give an idea of how the model behaves, they do not offer a comprehensive

understanding of the role of each feature in the prediction; XAI techniques must come into play [8, 9]. Furthermore, the development of explainability strategies for classical machine learning models can also benefit deep learning, as the techniques can be extended and adapted to suit neural networks.

After having analyzed the current scenario of biomedical deep learning, which this thesis proposes to render explainable, we will present effective XAI methodologies that allow the trustworthy usage of neural networks in biomedical applications, from disease gene discovery to drug development. When considering biomedicine and the usage of machine and deep learning models, we can think of a pipeline that, starting from databases containing genetic, molecular, and chemical information, allows researchers to devise models to a) discover genes involved in a disease's mechanisms, b) use the insights obtained to reuse known drugs for therapeutic purposes on the disease, and c) develop new treatments starting from what was discovered in the previous steps of the pipeline. If deep learning is used, its results are only satisfactory if they can be explained. For instance, a deep learning model suggesting a drug for a new treatment must also explain why the drug was chosen (e.g., because of the presence of some specific properties or features). Thus, XAI can help increase trust in predictive and diagnostic models, which otherwise would remain obscure and not usable in biomedicine. At the same time, explainability can aid in the extraction of new knowledge, for instance, by determining important features driving model predictions, leading to insights that are unobtainable when using a model in a black-box fashion. With this spirit, the work proposed in this thesis is meant to render explainable the deep learning components of the pipeline depicted in Figure 1.1. We will now describe it, introducing our solutions for each of its elements.

We can identify some key components in the proposed explainable biomedical deep learning pipeline. The first component pertains to the training of a gene discovery model (block 1). Disease gene discovery can be carried out in the context of network-based gene–disease association (GDA) identification or genome-wide association studies (GWAS). GDAs signify established associations between genes and diseases, indicating the involvement of specific genes in disease etiology and mechanisms. Bioinformatics and network medicine approach the problem via statistical, combinatorial [10, 11], or graph-based strategies [12]. Such methods rely on data derived from protein–protein interaction (PPI) and gene–disease networks, for which among the relevant databases we find BioGRID [13] and DisGeNET [14], respectively. Nonetheless, more recently, machine learning and deep learning-based approaches have emerged [15]. In contrast, GWAS [16] utilize patients' complete DNA set to ascertain whether a gene mutation's presence correlates with a disease.

Typically, this analysis is conducted with statistical instruments [17] using large-scale databases, such as UK Biobank [18], containing genetic information from thousands of individuals. After validation, the genes identified through these studies can be incorporated into the databases employed in gene–disease association studies. GWAS and network-based gene prediction share a common objective, adopting distinct data types and methodologies.

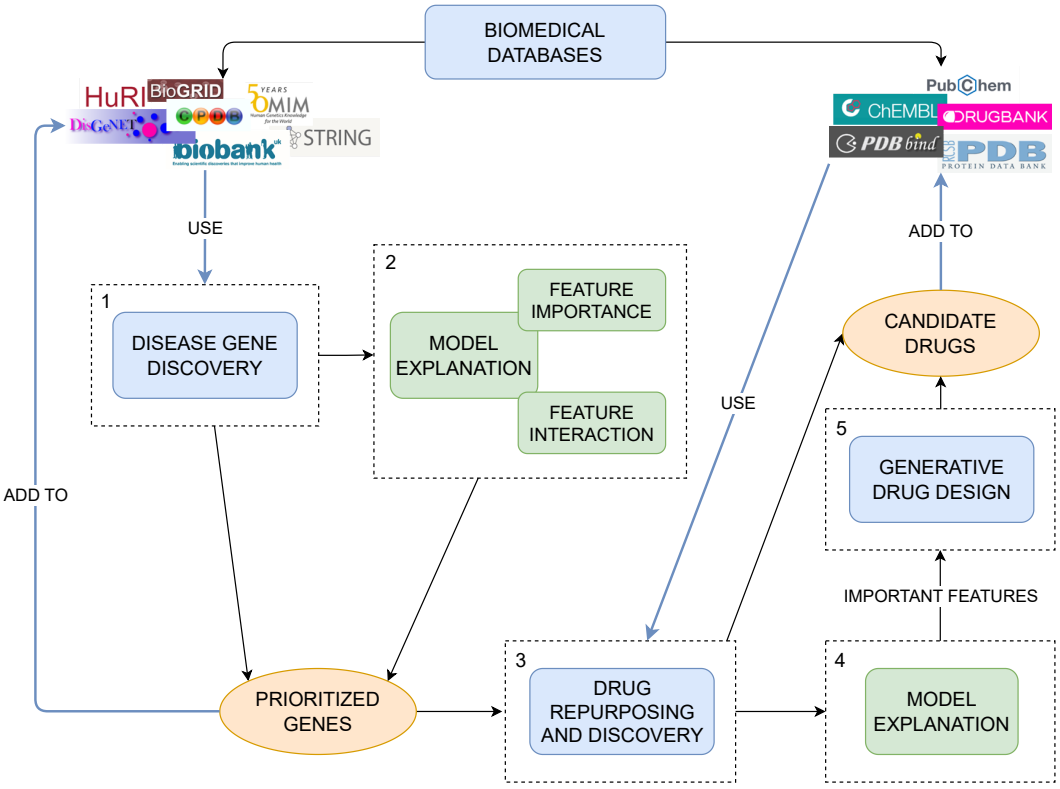


Figure 1.1. The proposed explainable biomedical deep learning pipeline. Each block of the pipeline finds a place within this thesis. Databases containing genetic information are used to train models for disease gene discovery (block 1). Those models are then explained, uncovering important features for predictions and possible interactions (block 2). The genes discovered in blocks 1 and 2 are integrated into the databases, augmenting the knowledge in the field. Such genes are then used as drug targets to find treatments for diseases (block 3), also exploiting information from chemical repositories. Block 4 pertains to the explanation of the drug discovery models. The important features found can drive the generative development of novel drugs (block 5). The drugs identified in blocks 3 and 5 are finally added to chemical drug–target databases, closing the pipeline.

When utilizing machine and deep learning to predict gene–disease associations, it is crucial to obtain explainable outcomes (block 2). In this scenario, two facets of explainability can be explored: feature importance and feature interaction. As the

name implies, the former aims to identify the influence of individual features on the prediction (e.g., the presence of a genetic mutation causing a trait). The latter concentrates on unveiling potential interactions among input features (e.g., whether two genes have a joint contribution to the likelihood of a disease). In this thesis, we will address both forms of explainability for neural networks within the domain of disease gene prediction.

Regarding this, we devised a methodology for feature importance to facilitate disease gene discovery by leveraging biological network data. The proposed method, XGDAG [19], detailed in Section 3.2, employs graph neural networks (GNNs) [20] for predicting associated genes while incorporating XAI techniques. This combination not only yields explainable outputs but also enables the identification of new gene associations in a novel utilization of the explainability concept. However, before explanation, a deep learning model must undergo appropriate training. Training in the field of bioinformatics poses challenges due to imbalanced datasets often characterized by a positive–unlabeled (PU) structure [21]. To address this issue, we have first devised a methodology for learning in PU settings, called NIAPU [22] and presented in Section 3.1. This serves as the foundation for developing our explainability strategy, as it enables proper learning for neural network models.

With regard to feature interaction, we will present EpiDetect, a comprehensive solution for addressing it within the context of epistatic interaction detection from GWAS data. Epistatic interactions [23] denote the nonlinear effect that two or more genetic variants, resulting from modifications occurring on a gene, exert on a phenotype or disease. Capturing these interactions is a nontrivial task since it involves examining the effects of all possible combinations of gene mutations across the DNA set. We relied on prior GWAS studies that narrowed down the potential genes implicated in these interactions to address this complexity. Subsequently, we employed a neural network to predict the occurrence of a disease. Lastly, as a main contribution, we developed a strategy for neural network explainability to reveal interactions among input genetic variants. We will present this work in Section 3.3.

Upon discovering disease-associated genes, it becomes possible to integrate them into the data sources utilized during their identification, thereby augmenting the knowledge pertaining to the investigated diseases and providing novel data for future research. This also opens the possibility to explore novel treatments. For example, it becomes feasible to repurpose existing drugs or design new ones to target the newly discovered associated genes or proteins (block 3). This thesis also addresses this issue in Section 3.4, considering primary biliary cholangitis (PBC) [24] as a case

study for drug repurposing. PBC is an autoimmune liver disease characterized by a lack of available treatments and was chosen as a target disease due to the significant interest in discovering potential cures for this condition.

Various approaches can address the challenges of drug repurposing and discovery. The approach we propose for PBC is bioinformatics-driven, where gene prioritization guides the identification of potential treatments. However, alternative chemoinformatics-based approaches can be employed with annexed XAI solutions (block 4). For example, machine learning or deep learning models can be trained to determine the activity of a given molecular compound against a specific target, using molecule structures or feature descriptors as input. Molecular activity prediction is a fundamental task in drug discovery, making it a relevant aspect of this thesis. Considering the graph-like nature of molecules, our chosen strategy involves training a graph neural network model to extract information from the molecular compounds and accurately predict their activity. We integrate explainability into our strategy, developing a novel approach tailored to GNNs. In Section 4.1, we introduce EdgeSHAPer [25], a new methodology that, first of its kind, approximates Shapley values from game theory [26] as indicators of edge importance, thereby identifying the most significant molecular substructures for the predictions.

Another paramount task in drug design, stickily related to molecular activity, involves compound potency prediction in protein–ligand interactions. This task entails evaluating the strength of the interaction between a molecular compound and a target protein. It can be approached as a regression task within machine learning. Given GNNs have been increasingly used in this domain, our research aimed to delve deeper into what GNNs truly learn when trained for such a purpose. GNNs leverage interaction graphs constructed from X-ray structures of protein–ligand interactions, from which three key components emerge: ligand compound, target protein, and intermolecular interactions. Existing literature suggests that ligand structures play a major role in potency prediction tasks [27] and no real learning of interactions occurs. Thus, our objective was to investigate more on this claim, utilizing the XAI technique proposed in Section 4.1. The obtained results, presented in Section 4.2, yielded unexpected findings and intriguing insights [28].

The studies above use our newly developed explainability strategy based on the approximation of Shapley values. However, it is worth noting that approximated Shapley values do not always yield satisfactory results. Notably, in the case of molecular activity prediction using SVMs, evidence suggests that approximation fails to accurately capture the true importance of molecular features [9]. With this

in mind, we aimed to address this limitation and propose a technique for the exact computation of Shapley values. Since SVM models are widely used in chemoinformatics, developing XAI methods for these has a high scientific relevance. The SVERAD method [29], proposed in Section 4.3, was effective in both computing Shapley values and reducing the expected running time of these calculations. At the same time, working with SVMs allowed us to ease the complexity associated with neural networks, laying the foundations for extending the exact computation of Shapley values to deep learning models in future research endeavors (Chapter 5).

Identifying important features driving predictions serves not only as a means to comprehend and rationalize model behavior but also as a starting point for developing new chemical compounds for drugs. This lays the basis of generative drug design (block 5). While not explicitly examined in this thesis, we will point to this aspect for future research perspectives. Generative artificial intelligence has recently emerged as a breakthrough in various domains, including drug discovery [30, 31]. This field can benefit from the incorporation of generative models into its pipeline. From generative adversarial networks [32, 33] to the advancements in large language models [34, 35], generative deep learning tools can create new molecular structures based on existing compounds and desirable drug properties. By combining this capability with our proposed XAI approaches, it becomes possible to infuse expert knowledge into these models, guiding the generation of novel drugs based on the prioritized important features identified during the preceding steps of the pipeline. Once the efficacy of the generated drugs has been validated, they can be integrated into existing databases, thereby fueling future research and filling the last gap in the explainable biomedical deep learning pipeline.

In light of the work it will present, this thesis is structured as follows. Chapter 2 will introduce the background and review the relevant literature on the main topics addressed. Chapter 3 will focus on the bioinformatics aspects of the proposed explainable biomedical deep learning pipeline, while Chapter 4 will analyze the chemoinformatics components. Lastly, Chapter 5 will draw the conclusions and discuss the central messages of this thesis while looking forward to future research directions.

Chapter 2

Background and Related Work

This chapter illustrates the background knowledge and related work of the main topics covered by the research, describing the areas involved in the proposed pipeline. We describe the challenges and how they have been tackled in the literature. This chapter is meant to give the reader the necessary background to understand better the proposed methods and the applications described in this thesis. As introduced, the topics analyzed cover two interconnected macro-areas of the broader biomedicine field: a) bioinformatics and network medicine, and b) chemoinformatics and medicinal chemistry.

2.1 Bioinformatics and Network Medicine

The first macro-area of biomedicine that was dealt with in this thesis is bioinformatics. This highly interdisciplinary field aims to investigate biological phenomena by leveraging the strengths of life sciences, computer science, and engineering, thereby creating specialized algorithms and methodologies tailored for analyzing biological data.

Bioinformatics relies on the use of *omics* data [36, 37], which pertains to large-scale biological datasets capturing comprehensive information about molecules and their interactions within biological systems. Encompassing diverse fields such as genomics, transcriptomics, proteomics, metabolomics, and epigenomics, each omics discipline focuses on distinct types of molecules, offering valuable insights into the structure, function, and regulation of biological systems.

In particular, genomics entails the study of an organism's complete set of genes, encompassing their sequences, organization, and variations. It offers a comprehensive perspective on an individual's genetic blueprint. Genomic data are typically

generated through techniques such as DNA sequencing [38], enabling the identification of genetic variations associated with diseases, including single-nucleotide polymorphisms (SNPs). The latter are substitutions of single nucleotides in a given position in the genome (called locus). Nucleotides, which are adenine (A), cytosine (C), guanine (G), and thymine (T), are the basic building blocks of DNA, forming genes. Nucleotide substitutions may be involved in the mechanisms of diseases or determine specific traits. SNPs are of particular interest in this thesis, and they will be analyzed in Section 3.3.

Transcriptomics focuses on transcriptome analysis, which is the complete set of RNA transcripts produced by the genome. There are various types of RNA. Messenger RNA (mRNA) plays a critical role in creating proteins. In this process, mRNA is transcribed from genes. Next, the mRNA transcripts are delivered to ribosomes in the cell cytoplasm. The ribosomes translate the mRNA and assemble amino acids into proteins. However, since not all genes code for proteins, not all RNA transcripts do. In fact, other types of transcripts are tasked with influencing cell structure and regulating genes. Transcriptomics provides insights into gene expression and can help researchers understand how genes are regulated. Techniques such as RNA sequencing [39] are commonly used to generate transcriptomic data.

Proteomics involves the investigation of the entire set of proteins present within a cell, tissue, or organism, commonly referred to as the proteome. Proteomic data deliver valuable insights into protein presence, modifications, and functional roles. Lately, there has been an increasing interest in studying interactions between molecules in cells, especially proteins, forming the so-called interactome. Interactomics deals with studying and analyzing the interactions between proteins and among proteins and other molecules in cells. These interactions form networks, like protein–protein interaction (PPI) networks and gene regulatory networks. This led to the birth of network medicine [12], a multidisciplinary science that leverages the knowledge embedded in network representation of biological data to extract novel insights.

Additional omics data are represented by metabolomics and epigenomics. The former centers on the analysis of metabolites, molecules offering valuable insights into metabolic pathways and biochemical processes. The latter explores modifications to genes and their associated proteins that can influence gene expression and cellular function. Epigenomics aims to understand gene expression regulation and the impact of epigenetic changes on development, aging, and diseases.

Bioinformatics and network medicine harness the representative power of omics data

and, using algorithms and computational strategies that go from combinatorial to machine and deep learning-based approaches, aim to gather insights into biological systems and enable researchers to uncover complex interactions and disease mechanisms. Consequently, particularly effective are multi-omics approaches [40], which combine information from different omics data sources in the creation of multiplex networks [41], able to capture the knowledge embedded in the interactions at (and between) different omics levels, trying to mimic the complexity of biological organisms.

In this thesis, we made use mainly of interactomics and genomic data: the former as gene–disease and PPI networks (Sections 3.1, 3.2, and 3.4) and the latter in the form of DNA sequences and genetic mutations (Section 3.3). This represents the first part of the explainable biomedical deep learning pipeline, whose task is to look for disease-associated genes and interactions between them.

2.1.1 Gene–Disease Associations

In particular, discovering disease-associated genes is one of the central tasks addressed in bioinformatics and network medicine. This refers to the first important block in our pipeline. Gene–disease associations (GDAs) refer to the connections or relationships between specific genes and the occurrence or development of certain diseases or medical conditions. GDAs are critical for understanding disease etiology and tailoring effective interventions and treatments.

GDAs are established through various approaches, including genome-wide association studies (GWAS) [42], linkage analysis [43], and genetic sequencing [44]. Given the great computational power modern systems offer, GWAS have become particularly popular in recent years. They involve analyzing the genetic variations across large populations to identify common genetic markers or variants associated with specific diseases. Differently, linkage analysis examines the inheritance pattern of diseases within families to pinpoint genome regions that may contain disease-causing genes. Genetic sequencing technologies have also advanced significantly, allowing researchers to identify disease-causing mutations in specific genes.

Once GDAs are identified, they can have several practical applications. They can aid in disease prediction and risk assessment by determining if an individual carries genetic variations that increase their susceptibility to certain diseases. Furthermore, GDAs can inform personalized medicine by guiding therapy decisions and drug treatments (as discussed in Section 3.4). Targeted therapies can be developed to

modulate the activity of disease-associated genes or proteins, potentially leading to more effective cures.

The knowledge about GDAs gathered through the years gave birth to large databases, free to use by the research community. Paring this knowledge on disease-associated genes with insights from the network-like representation of biological phenomena (the interactome), promising results in GDA discovery are coming from network medicine approaches [12, 45] leveraging network data, such as PPI networks. Among the most used PPIs, we find BioGRID [13], HuRI [46], and STRING [47]. In these networks, nodes are proteins (or protein-coding genes) that are connected if an interaction exists. The rationale behind using those data is that interaction mechanisms between proteins can determine the pathogenesis of diseases. Thus, it is possible to study those networks to find genes whose mutation may generate proteins involved in disease mechanisms.

For gene discovery purposes, these interaction networks are extended with information on disease associations, for which databases such as DisGeNET [48, 14] and eDGAR [49] are typically used. Many gene detection techniques that rely on those data have been developed. Among the most known approaches are DIAMOnD [50] and DiaBLE [51], which rely on the concept of *connectivity significance* for finding new candidate disease genes. This concept states that genes connected to associated genes in an interaction network are more likely to be associated genes themselves. Starting from a set of associated genes (seed genes), DIAMOnD assigns a value to each neighbor based on its connectivity significance and determines a gene ranking. In this way, it is possible to evaluate genes that have more connections to known seed genes than expected. Using a statistical test, the most significant gene is considered a putative gene and added to the set of seed genes. The discovery process is repeated until the number of desired genes is retrieved. DiaBLE uses the same rationale as DIAMOnD but introduces a new variant of the connectivity significance score, relying on an adaptive gene universe for the statistical test. Instead of using the whole interactome, as in DIAMOnD, it considers a smaller set of genes that expands at any iteration as more candidates are found. This improved the retrieval performances of associated genes compared to DIAMOnD [51].

Other gene discovery techniques, such as DOMINO [52], use machine learning to determine associated genes. Another approach, Markov clustering [53, 54], is based on the simulation of stochastic flow in graphs. Based on such principle, clusters are created, and the presence of known disease genes in a cluster makes the other elements putative disease genes. Another line of work uses random walks with restart [55, 56]

for gene discovery. Following the guilt-by-association strategy [57] they explore the PPI network vicinity of known disease genes based on the premise that nodes related to similar functions tend to lie close to each other. GUILD [58] is based on the hypothesis that the interconnections among disease genes in the interactome are captured by taking into account the relevance of the paths connecting the disease genes, using different topology-based ranking algorithms. ToppGene [59, 60, 61] exploits a fuzzy-based similarity measure to compute the similarity between any two genes based on semantic annotations, and ToppNet [61] uses extended versions of the PageRank [62] and HITS [63] algorithms, applied over the interactome topology.

Furthermore, gene discovery can be framed as a positive–unlabeled (PU) learning problem [21]. Differently from classic machine learning scenarios, in which a binary dataset consists of positive and negative samples, in PU learning, instead of negative samples, we have a set of unlabeled instances, which can be regarded as a set of negative elements and some positive samples that have not yet been discovered. A machine learning-based method for gene discovery working in a PU setting is ProDiGe [64]. It uses biased begging support vector machines (SVMs) [65] trained on positive and unlabeled instances. Training in a binary PU setting can impinge on the quality of the model due to the noise introduced by unlabeled positive samples treated as negative [66]. For this reason, different PU learning strategies approach gene discovery using two-step techniques, such as PUDI [67] and EPU [68]. Those techniques work by first identifying a set of highly likely negative genes and then training a classifier. Pseudo-classes with different likelihoods of *positiveness* can be introduced to ease the learning and give meaning to the unlabeled instances. Introducing pseudo-classes enables proper learning for machine and deep learning models to be used as gene discovery tools. In fact, training machine learning models, in particular neural networks, in this scenario is made hard by the extremely unbalancing of the dataset and the fact that the unlabeled samples may contain possible positive genes. To cope with this, we propose a technique called NIAPU [22] (Section 3.1) that assigns pseudo-labels to the unlabeled elements, enabling proper learning. Then, we use this technique to train a graph neural network (GNN) as the basis for XGDAG [19], our explainable artificial intelligence (XAI)-based gene discovery tool (Section 3.2). Finally, we successfully apply the NIAPU pipeline for GDA-driven drug repurposing [24], as shown in Section 3.4.

2.1.2 Epistatic Interactions

More complex than single gene–disease associations, two genes may interact, influencing a trait or disease in a nonlinear manner. This mechanism is known as epistatic

interaction [23] (or epistasis) and refers to the phenomenon where the effect of one allele is modified by other alleles. An allele is a version of a gene or DNA sequence at a given locus. An individual inherits two alleles, one from each parent, which will determine the genotype, which can be homozygous (same allele) or heterozygous (different alleles). In epistasis, interactions between genes influence the expression of a phenotype in a way that cannot be predicted solely based on the individual effects of each gene. It is a common occurrence in genetics and can have significant implications for understanding the genetic basis of complex traits and diseases that are not only governed by one gene's mutation.

There are different types of epistatic interactions [69], including positive epistasis and negative epistasis. Positive epistasis occurs when the combined effect of two or more genes is greater than the sum of their individual (marginal) effects. In this case, the presence of one gene nonlinearly enhances the effect of another gene. It can contribute to genetic buffering, where genetic variations that individually have mild (or null) effects on a trait can produce a more noticeable impact when combined or even create new phenotypes. Negative epistasis, on the other hand, refers to situations where the effect of one gene masks or suppresses the effect of another gene. In this case, the presence of one gene attenuates or counteracts the expression of the other gene, resulting in a less pronounced (or absent) phenotype than expected based on the individual effects of each gene.

Recent advances in GWAS and the significant increase of available data have helped identify thousands of robustly associated loci and have provided novel insights about biological pathways underpinning complex diseases and traits. However, the extent to which potential epistatic interactions contribute to complex networks that dictate the molecular mechanism underlying phenotypes remains to be determined. Studying epistatic interactions can be challenging due to the large number of potential gene combinations and the complexity of the underlying genetic networks. Researchers have developed several approaches to examine epistatic interactions [70]. One of the most known and effective is Multifactor Dimensionality Reduction (MDR) [71]. It constructs a single attribute from variables, reducing data dimensionality. Genotypes at different loci are pooled into high- and low-risk groups. This reduces the dimensionality by creating a multilocus genotype variable, which can predict the disease status and assess the joint effect of the merged genotypes. Unfortunately, MDR suffers from a high computational cost due to the number of possible genotype combinations. Another approach, Boolean Operation-based Screening and Testing (BOOST) [72], relies on a logistic regression model that considers both marginal and pairwise interaction effects. Relying on a boolean representation of the data for

computational efficiency, this method can detect epistatic pairs quickly. Additional methods proposed for epistasis detection are based on random forests (RFs) [73, 74] and Bayesian networks [75].

A caveat of these approaches is that they are influenced significantly by marginal (or main) effects. A marginal effect is the effect a genetic variant alone has on a trait. This can confuse the models and make them believe that a merely additive phenomenon caused by two high marginal effects can be an epistatic interaction. Main effects can be tricky because they may lure algorithms into believing that a nonlinear interaction exists, whereas it may just be the product of two additive effects that act independently.

As described in Section 3.3, we addressed this problem by developing a neural network-based pipeline powered by an XAI component to rationalize the predictions. This method, called EpiDetect, opens the black box and finds interacting features that determine the output, detecting epistatic pairs and filtering out possible marginal effects.

2.2 Chemoinformatics and Medicinal Chemistry

After the discovery of genes associated with diseases, the second part of the explainable biomedical deep learning pipeline falls mainly in the domain of chemoinformatics [76, 77, 78]. This term dates back to 1998 and initially referred to the information resources needed to optimize a molecule to render it a drug. However, the more the field developed, the broader the definition became. Nowadays, it relates to the development and application of computational methods and tools for storing, organizing, retrieving, analyzing, and predicting chemical information, encompassing a broad spectrum of topics, from data representation to structure and property prediction and drug development. Chemoinformatics researchers use vast amounts of chemical data to extract meaningful insights and knowledge to facilitate decision-making processes and accelerate research and development in the chemical domain, especially for medicinal chemistry applications [79].

Consequently, one of the primary areas of focus is the representation and encoding of chemical structures and properties to be handled by computer programs. For this purpose, chemical structures are typically represented using molecular graphs or simplified line notations, such as the widely used Simplified Molecular Input Line Entry System (SMILES) [80] or International Chemical Identifier (InChI) [81]. These representations enable efficient storage, search, and retrieval of chemical information

from databases for subsequent analysis with statistical and machine learning methods.

For medicinal chemistry research, chemoinformatics encompasses the development of computational algorithms and models for analyzing and predicting chemical properties and activities. Quantitative structure–activity relationship (QSAR) [82] models, for example, correlate chemical structures with their biological or physico-chemical properties, allowing researchers to predict the behavior of new compounds based on existing data. Other approaches include molecular docking [83], molecular dynamics simulations [84], and machine learning techniques to understand and predict chemical interactions, binding affinities, and molecular behavior [85]. In this regard, the relatively recent advent of more complex and accurate machine learning models, including neural networks-based approaches that are emerging as leading strategies [86, 87], increased the opportunities offered by computational methods in medicinal chemistry research [88].

2.2.1 Drug Discovery

The cornerstone of chemoinformatics is represented by its applicability in drug discovery [89, 90]. This is the first and highly interdisciplinary process in the drug development pipeline, through which new candidate drugs are designed and proposed for new treatments, to later undergo preclinical and clinical trials for final approval. Computational methods have lately become fundamental to speed up and optimize the process. The drug discovery pipeline is composed of several steps.

First, a potential target needs to be found. This target can be either a protein, an enzyme, a gene, an RNA fragment, or a biological pathway that plays a key role in the mechanisms of the investigated disease. As described previously, the candidate target can be identified through bioinformatics-based strategies, such as network-based approaches [91] or GWAS studies [92] for gene–disease association discovery. The methods proposed in this thesis, NIAPU, XGDAG, and EpiDetect, can be used for this purpose.

After identifying a target, several screening approaches are used to find a suitable *hit* compound. Hit compounds are molecules able to interact and bind with the chosen target. A first approach, called high-throughput screening (HTS) [93, 94, 95], aims at rapidly identify compounds that can modulate the target of interest, heavily relying on liquid-handling robots and automation processes to test in parallel large numbers of molecules [96]. HTS can also be used for toxicity assays [97, 98]. Another solution for hit compound identification is offered by virtual screening [99]. It involves using

computational methods, including deep learning-based strategies [100], to screen large databases of chemical compounds and suggest those with potential therapeutic activity against the biological target. The selected compounds undergo subsequent experimental validation. This approach helps perform molecular activity prediction and assess toxicity, prioritizing compounds in a cost-effective manner saving time and resources in the drug development pipeline. It is also possible to predict the ADME properties (absorption, distribution, metabolism, and excretion) [101], which describe how a drug is processed by the organism, of critical importance for its effective use. Virtual screening is used to select promising hit compounds by leveraging libraries of chemical structures. Still, it also proves to be a powerful tool for drug repurposing applications [102, 103, 104], screening against databases of already available and approved drugs.

In this phase, compound activity and potency prediction are central tasks for which a variety of computational methods are employed. The research community made substantial efforts in developing computational methods to discover active compounds and perform fast and accurate virtual screening. Those models can help determine if a molecule is a hit compound or not, as a classification problem, or predict its potency in the interaction, as a regression problem. Molecular activity and potency prediction are two related and interconnected tasks in drug design. Ligand-based studies rely on ligand molecule information to predict binding activity or affinity. Differently, structure-based studies leverage the structural information of the drug target, such as the binding sites. Both ligand and protein information can be exploited, either independently (pair-based studies) or explicitly considering the interaction between them (the so-called protein–ligand complex) in complex-based studies.

In ligand-based design, computational methods include multiple linear regression models for QSAR modeling [105, 106], SVM models, or RF regression [107, 108]. In addition, deep learning is increasingly employed [85, 109, 110, 111]. In structure-based design, ligand binding energies are roughly approximated using different types of molecular mechanics-based scoring functions [112, 113, 114] or predicted using quantum mechanics/molecular mechanics (QM/MM) [115] and alchemical relative free energy perturbation calculations [116]. Similar to the situation with ligand-based predictions, the increasing popularity of deep learning has also triggered neural network applications in structure-based design, such as the use of convolutional neural networks to predict ligand affinity from voxel representations of ligand binding sites [117, 118]. Furthermore, GNNs including message-passing neural networks (MPNNs) [119] have recently been investigated for the prediction

of affinity from protein–ligand interaction graphs [27], which are usually obtained from X-ray structures of protein–ligand complexes. We will now describe in more detail some salient state-of-the-art machine and deep learning solutions for molecular activity and binding affinity prediction.

Classic machine learning algorithms usually work with predefined features, such as molecular or interaction fingerprints [120, 121, 122], among which we find the widely used Morgan fingerprints [123]. Fingerprints are binary vectors describing whether a molecular feature is present or not. Differently, deep learning models mostly leverage molecular structures, either in the form of 3D grids or explicitly working on graph representations. Machine learning algorithms, such as SVMs and RFs, have been largely employed for compound activity prediction and established themselves as mainstays in molecular machine learning, with many successful applications and investigations [108, 124, 125, 126, 127, 128, 129, 130].

Focusing on deep learning approaches, simple multilayer perceptrons (MLPs) [131] using fingerprints proved effective for classifying active molecules [132] and looking for inhibitor compounds [133]. However, the most effective deep learning solutions employ molecular or interaction structures, either in terms of 3D grids or graph representations. Among the first strategies of this kind, we find AtomNet [134], a convolutional neural network (CNN)-based model that operates on 3D grids constructed out of protein–ligand complexes to classify active compounds. This work was the beginning of a series of successful CNN-based solutions for molecular activity prediction. Indeed, in the same family of strategies, we find BindScope [135] and Pafnucy [118], both employing as input voxelized representation of ligand and protein complex structures for compound activity and potency prediction, respectively. DeepDTA [136] uses two CNNs applied to ligand and protein representation separately, later concatenated in a combined representation for the final binding affinity regression. This method was later extended to WideDTA [137], taking as input ligand substructures information and protein motifs. DeepAtom [138] was later developed and, employing the same rationale as the previous complex-based approaches, outperformed its counterparts in potency prediction. Another convolution-based method is DeepConv-DTI [139], which uses Morgan fingerprints [123] for ligand representation and raw protein sequences for activity prediction. Differently, recurrent neural networks (RNNs) have also been employed. For instance, DeepAffinity [140] is an autoencoder model unifying RNNs and CNNs in an effective strategy that, taking as input protein as aminoacid sequences and ligand molecules as SMILES representations, predicts the binding affinity of the complex.

After the success of convolutional and recurrent models, GNNs have started to be explored in molecular activity prediction endeavors in drug design [141]. Graph structures of molecules or ligand–protein complexes are used as input to those models since GNNs can powerfully leverage network data representations and generate embeddings for nodes and edges that can be used to derive predictions. Nodes can contain features describing atomic properties and edges can represent intramolecular or intermolecular bonds. For instance, GraphBAR [142] is a GNN-based strategy that uses graph convolution applied to interaction graphs constructed from binding pockets to predict the affinity value. Furthermore, GNNs with distance-aware graph attention mechanisms [143] outperform previously developed CNN-based models, such as AtomNet. Moreover, PotentialNet [144] leverages a gated graph attention network to determine binding affinity using graph representations of protein–ligand complexes aggregating information at different stages (from connectivity to spatial information). Mixed models were also developed, for instance by merging RNNs with graph convolution [145]. Pair-based strategies using GNNs were also particularly effective. Graph convolution was applied to independent graph representations of ligands and protein pockets, whose neural network embeddings were subsequently merged for final binding classification [146], outperforming results obtained with classic machine learning strategies and 3D convolution approaches. DGraphDTA uses a similar strategy for affinity prediction [147]. Finally, MPNNs were used for binding affinity prediction, employing graph representation of proteins, ligands, and interaction complexes [27], also used in cascade with ARMA-based GNNs [148, 149].

One enormous advantage of deep learning is that it does not need predefined features such as fingerprints. Conversely, the learning process extracts meaningful features from input representations, like grids or graphs. The setback is, as we already discussed, the lack of interpretability of neural networks. In this thesis, given the promising results and the possibilities that GNNs offer, we decided to exploit their representation power for the topical task of molecular activity prediction, pairing them with a newly developed XAI strategy called EdgeSHAPer [25] (described in Section 4.1). Initially created for activity classification, we extended this methodology to potency regression, to investigate GNN applicability to the strictly related task of protein–ligand affinity prediction and determine what GNNs truly learn when applied to interaction graphs obtained from protein–ligand complexes [28] (Section 4.2).

After finding the most promising hit compounds, they are refined to produce more potent (with high binding affinity) and selective (avoiding undesirable targets) compounds to be used with in vivo models: lead compounds are thus identified. This is the hit-to-lead phase. The final drug design step is lead optimization [89, 150].

It consists of modifying the molecular structure to reduce the deficits of the lead compound, reducing the risk of adverse effects while, at the same time, maintaining all the desirable drug properties. After optimization, a lead compound is selected as a candidate drug for preclinical and clinical testing.

2.2.2 Drug Repurposing

Differently from de novo drug design, drug repurposing [151] consists of repositioning existing drugs beyond their original therapeutic target as a cure for untreated diseases. This thesis treats this task as a bridge between bioinformatics and chemoinformatics since it can be tackled with both network-based and virtual screening-based strategies, sometimes with evident overlaps between the methodologies that blur any clear distinction between the areas. By leveraging the vast knowledge of already approved drugs, it is possible to address unmet clinical needs more safely and cost-effectively, capitalizing on drugs that have undergone preclinical and clinical tests, including toxicity studies.

Computational drug repurposing can employ different strategies [152]. Genome-based strategies, for instance, can leverage the knowledge about gene–disease associations and gene expression profiles (the activity of genes). The Connectivity Map (CMap) [153] is a key resource for computational drug repositioning endeavors. It contains genome-wide expression data that can be exploited to identify associations between genes related to a specific disease or targeted by a specific drug. This resource has been extensively used in cancer research with interesting results [154, 155]. Differently from genome-focused drug repurposing, usually exploited by bioinformatics approaches, chemical structure-based strategies use molecular information assuming that drugs sharing similar structural properties may modulate genes or proteins analogously; studying the chemical similarity between drugs led to the discovery of unknown drug–target associations over the years [156]. Chemoinformatics approaches using chemical structure information, both in terms of molecular fingerprints or three-dimensional information [157], have been extensively used for drug repurposing applications [152]. A third family of strategies is the phenome-based one. The phenome is the set of all phenotype information. This strategy is based on the hypothesis that if two diseases share similar phenotypic profiles and a drug is used to cure one disease, the same drug may be proposed as a treatment for the other disease [158], also using side-effect similarities to understand if two drugs share the same target [159].

Network-based strategies have been demonstrated to be effective among the compu-

tational approaches to drug repurposing. Those solutions use biological networks as input, enriched with drug information. Those networks may contain nodes representing genes, proteins, phenotypes, or additional entities connected according to different criteria. These approaches can give insights into how drug targets work, thus discovering new cures for untreated conditions. We find many successful network analysis solutions for drug repurposing in the literature. For instance, by combining PPI networks with drug–target interaction networks in a bipartite graph, it is possible to predict interactions between a target protein and a drug [160]. Another approach [161] employs drug–drug interaction networks for polypharmacology side effects using chemical structure information and drug–target similarity. This is an example of hybrid bioinformatics and chemoinformatics methodology since it exploits both network science and molecular information. Additional network-based approaches use clustering techniques on top of drug–disease networks [162] built by using GDAs and drug–target networks to identify modules containing possible candidate drugs for repurposing. Gene similarity and chemical structure information can be used jointly to build drug–drug interactions in a hybrid approach [163], and information from heterogeneous data sources can be exploited by network-based prioritization models with compelling results [164]. Finally, network centrality measures can be employed to find potential drugs using networks built from molecular data, drug–drug interactions, and additional sources [165, 166]. Lately, knowledge graphs [167, 168] have gathered attention given the huge amount of information they contain and their promising results [169, 170, 171]. The drug repurposing strategy we will present in Section 3.4 is a network-based approach, and this is why it finds its place in Chapter 3, dedicated to bioinformatics and network medicine. It works by finding new disease-associated genes and proposing drugs targeting such genes as candidate treatments.

Machine and deep learning are also extensively used for drug repositioning to unveil unknown interactions between old drugs and novel targets. In this, approaches similar to the ones used for compound activity and potency prediction can also be employed since the tasks share some major common points, like the need for the drug to be active against the new target. More in detail, classic machine learning approaches, such as logistic regression leveraging chemical structures, side effects, and disease–disease similarities, have been used [172]. RF models proved effective for virtual drug screening in personalized medicine by using patient-specific genomic traits [173]. Moreover, SVMs have been employed using a drug similarity matrix-based kernel [174]. Additional effective machine learning attempts are represented by causal inference-matrix factorization [175], collaborative filtering [176], and recommendation systems [177]. The explosion of deep learning also hit drug

repurposing. Paired with the extensive growth of the data available, needed to train neural networks properly, many different accurate algorithms have been developed. The ability of neural networks to learn the nonlinear relationships between input features renders them suitable for learning associations between drugs and biological entities in complex systems [88]. MLPs were used to predict drug properties and find potential new indications for old drugs [178]. Leveraging data from gene expression profiles and biological pathways, such models outperformed classic machine learning techniques like SVMs. Furthermore, one-shot learning strategies integrating CNNs [179] and long short-term memory (LSTM) [180] aided in the training of deep learning models when small amounts of data are available [181]. Another work leverages CNNs taking as input drug structures and protein sequences to determine interactions between drugs and target proteins [182]. Moreover, variational autoencoders [183] have been demonstrated to be powerful solutions to infer new candidate drugs learning from drug databases, such as drug–drug, drug–disease, and drug–target associations networks [184]. More recently, the impactful research concerning graph neural networks [20] and the inherent graph-like structures of molecules, from drugs to proteins, led to the development of deep learning solutions based on GNNs. A method called Decagon [185], which is based on graph convolutional networks (GCN) [186], can unveil side effects due to drug–drug interaction, essential knowledge when repurposing drugs for novel treatments. GDRnet [187] is a GNN that deals with drug repurposing as a link prediction problem: by working on a graph interconnecting genes, drugs, diseases, and anatomy information, it can predict novel drug–disease links.

Drug repurposing is particularly useful in time-sensitive scenarios where using an old drug is faster and safer than designing a new one. Notably, this has been of utmost importance in the pursuit of therapies for COVID-19 [188, 189], in which computational methodologies found their place [190, 169, 191, 170, 192, 193].

Many solutions present in the literature merge the realms of bioinformatics and chemoinformatics since those techniques, deep learning in particular, use multimodal input data that go from genomics to chemical structure and biological networks coalesced into a single complex knowledge base. Such complexity makes it hard to rationalize the predictions made by deep learning models: none of the methods mentioned can derive interpretable results. As extensively remarked, in any life science-applied context, the black-box character of deep learning models limits their effectiveness and trustworthiness. XAI solutions, like the already mentioned EdgeSHAPer, need to be devised to rationalize the choices of the models and enable their practical use in the drug development pipeline. Moreover, taking a step

back from neural networks and given their consistent importance in pharmaceutical research, we also propose an explainability strategy for SVMs called SVERAD [29], which will be described in Section 4.3. The proposed XAI solutions find their application in both de novo design and drug repurposing. When new drug–disease links are established, the findings can be integrated into chemical databases at the service of future medicinal chemistry research.

2.3 The Need for Explainability

The increasing popularity of neural network architectures across many areas of science, including bioinformatics and chemoinformatics, has raised pros and cons. On the one hand, deep learning has led to unprecedented progress in areas such as computer vision, natural language processing, and network analysis and has opened the door to new scientific applications going beyond the capacity of standard machine learning. On the other hand, it has partly mystified machine learning and triggered high expectations concerning the problem-solving ability of machines and their putative ability to arrive at decisions beyond human reasoning. A characteristic feature of most machine learning methods, by no means confined to neural networks, is their often quoted black-box character [194], meaning that the decisions of those models remain machine-internal and are extremely hard to comprehend. The black box issue has been on the machine learning agenda for decades, working against the acceptance of machine learning results to guide experimental design in many areas, including bioinformatics and medicinal chemistry [129, 195]. With increasingly complex neural network architectures being employed for many scientific applications, the problem has further increased in magnitude. In interdisciplinary research settings in life sciences, including disease gene discovery and drug design, the natural reluctance of experimentalists to rely on machine learning results that they cannot rationalize in biological or chemical terms often limits the impact of machine learning research in the field [129, 195]. Such limitations are being recognized and, as a consequence, with the advent of deep learning, there are increasing discussions in the field about the relationship between model complexity and interpretability and the tendency to use models that are too complicated for prediction tasks at hand [196], which would need to be backed up by explanation methods to be trusted [197]. Thus, increasing attention is paid to explainable machine learning [198, 129] and the overarching area of XAI [5, 199, 200, 201]. XAI refers to different categories of computational approaches for rationalizing models and their decisions in different areas of basic and applied research [5, 200, 201]. Importantly, XAI approaches should not only help domain experts rationalize predictions, but model explanations should also be accessible to non-expert investigators in interdisciplinary settings [202, 129]. Expla-

nation methods are equally relevant for classification and regression models [203, 204]. Conceptually different XAI approaches include methods for feature weighting or attribution [8, 205], causal methods [206], counterfactual and contrastive explanations [207, 208], transparent probabilistic models, or graph convolution analysis methods [199, 200]. While interest in XAI is steadily increasing, the field is far from being mature, and relevant approaches are often still in early exploratory stages, especially in life sciences [85, 200]. This calls for in-depth research and further applications in the area.

The first attempts to explain neural network decisions date back to 1991 with Garson's algorithm [209], the first approach that tried to use neural network connection weights to find important input features. This led later to some improved variants [210, 211], and other weight-based approaches [212, 213]. Consequently, more sophisticated and accurate solutions were proposed, like computing partial derivatives of the output with respect to the input neurons as an importance metric [214, 215] or, more recently, taking into consideration the input gradients [216, 217] like in Integrated Gradients [218]. In the literature, we also find examples of explainability methods specific to certain types of neural networks. Convolutional neural networks can be studied via feature map visualization by visualizing the convolution filters [219] and saliency maps [220], and some works tried to understand LSTMs by extracting relationships between variable sequences [221]. Along with model-specific approaches, model-agnostic solutions are particularly sought after, given their general applicability to algorithms and models of varying complexity. One of the most used approaches is Local Interpretable Model-agnostic Explanations (LIME) [222]. It defines a local approximation model perturbing the input data to see how the prediction is affected to investigate the relationships between input and output variables, delivering instance-based explanations. The greatest success was obtained by Shapley value-based explainers [26], such as Shapley additive explanations (SHAP) [8]. The latter is a methodology that relies on game theory to determine feature importance, featuring also a neural network-specific variant, DeepSHAP, which is based in turn on another explainability method called DeepLIFT [205], which measures the impact of a feature by comparing it to some reference neutral value. Given their impact on XAI research, we will delve deeper into how Shapley values can be used for model explanation and the methodologies employing them. More rare and less explored in XAI is the study of feature interaction. Few approaches are available. Neural Interaction Detection (NID) [223] uses neural network weights for this purpose, while AutoInt [224] employs self-attention mechanisms to discover interacting features. As also aforementioned, in Section 3.3, we will approach this uncharted area of XAI by proposing a novel feature interaction detection method used in the context of

epistatic interactions.

2.3.1 Explaining Graph Neural Networks

When approaching graph neural network explainability, we dive into a different world. Although standard importance attribution methods can be used, they do not capture the essence of graphs. They can deliver explanations in terms of important features, but they don't include the building blocks of graphs: nodes and edges. In this regard, GNNExplainer [225] was the first XAI method designed explicitly for GNNs. It works by learning a mask on edges and node features via an optimization process. It obtains a subgraph that maximizes the mutual information between the GNN prediction and the distribution of possible subgraphs. Later, PGExplainer [226] was proposed as a generalization of GNNExplainer. It uses a neural network to learn the parameters to use in the explanation process, enabling it to work in inductive settings. It delivers edge-based explanations. In addition, a method termed GNN Explanation Supervision has been reported that combines node- and edge-based model explanation through graph regularization techniques, aiming to achieve consistency between node- and edge-based explanations through supervised adaptive learning [227]. For GCNs, edges important for model decisions have also been identified using previously introduced agnostic local explanation models [228]. In addition, MPNN variants with self-attention mechanisms have been reported to enable the extraction of learned attention weights [229, 230]. Furthermore, self-explainable GNNs are investigated that aim to identify k-nearest labeled nodes for each unlabeled node based on node and graph structural similarity to generate an explainable node classification [231]. Moreover, LIME was extended to graphs with GraphLIME [232]. It uses the Hilbert-Schmidt Independence Criterion Lasso [233] for nonlinear feature selection, but it only delivers explanations in terms of node features. Another methodology is PGM-Explainer [234]. This method looks for nodes that are crucial for the model by using an interpretable Bayesian network that approximates the real prediction. A different perspective is offered by XGNN [235]. It is a model-level explainer that uses reinforcement learning to train a generator to derive a graph that maximizes the model's prediction. The explanation is thus a complete graph that gives insights into the GNN model behavior. Finally, several methods leveraging Shapley values were also devised to explain GNN models. Due to the topic's relevance in this thesis, we will discuss those methodologies in a dedicated section.

2.3.2 The Shapley Value Concept

Shapley values were first introduced in cooperative game theory [26] to quantify individual players' contributions to a team's performance. In a cooperative game, players make an individual contribution toward a common goal (the team's reward, payout, or gain), working in coalition with each other. The Shapley value is a way to fairly distribute the payout among the players according to their contribution to the game; it represents the average marginal contribution of each player across all possible player coalitions. This concept can be translated from game theory to machine learning to explain model predictions [236]. In machine learning, the game is the prediction task, the players are input features, and the model's output represents the reward. This makes it possible to rationalize the model's decisions in terms of salient features, with the Shapley value representing the feature's marginal contribution to the output (thus, the feature's importance).

In order to compute the Shapley value for feature f, let $\mathcal{F}$ be the complete set of features and $\mathcal{S}$ a coalition of features (a subset of $\mathcal{F} \setminus \{f\}$). The contribution $\phi_f(v)$ is computed by considering the difference in the value v (the model's prediction) of the coalition $\mathcal{S}$ with and without the assessed feature f, weighted by the inverse multinomial coefficient $\binom{|\mathcal{F}|}{1,|\mathcal{S}|,|\mathcal{F}|-|\mathcal{S}|-1}^{-1}$, which is calculated as the number of permutations of the coalition ($|\mathcal{S}|$) multiplied by the number of features not contained in the coalition ($|\mathcal{F}| - |\mathcal{S}| - 1$) and divided by the number of all possible feature permutations ($|\mathcal{F}|!$). This must be repeated and summed for all possible subsets $\mathcal{S}$ of the $\mathcal{F} \setminus \{f\}$ features, obtaining the following equation:

$$\phi_f(v) = \sum_{\mathcal{S} \subseteq \mathcal{F} \setminus \{f\}} \frac{|\mathcal{S}|!\,(|\mathcal{F}| - |\mathcal{S}| - 1)!}{|\mathcal{F}|!} \left(v(\mathcal{S} \cup f) - v(\mathcal{S})\right). \tag{2.1}$$

Shapley values are the only feature attribution method that satisfies some important properties [237], namely *efficiency*, *symmetry*, *dummy* (or null player), and *linearity* (or additivity). Those properties guarantee a fair distribution of the payout.

Efficiency states that the sum of all feature contributions (the Shapley values) must add up to the difference between the prediction for input $\boldsymbol{x}$ and the average prediction over the data points $\boldsymbol{X}$ (expected value):

$$\sum_{i=0}^{|\mathcal{F}|-1} \phi_i(v) = v(\boldsymbol{x}) - E_X(v(\boldsymbol{X})). \tag{2.2}$$

According to the symmetry property, the contributions of two features i and j should be equal if they contribute equally to all possible coalitions, so, if $v(\mathcal{S} \cup \{i\}) = v(\mathcal{S} \cup \{j\})\ \forall \mathcal{S} \subseteq \mathcal{F} \setminus \{i, j\}$, then $\phi_i(v) = \phi_j(v)$.

The dummy property indicates that if a feature i has no impact on the value, regardless of which coalition of features it is added to, it should have a Shapley value of 0. So, if $v(\mathcal{S} \cup \{i\}) = v(\mathcal{S})\ \forall \mathcal{S} \subseteq \mathcal{F} \setminus \{i\}$, then $\phi_i(v) = 0$.

Finally, the linearity property states that for a game with combined value functions v and w, the Shapley values for feature i in v and w are such that

$$\phi_i\left(v + w\right) = \phi_i\left(v\right) + \phi_i\left(w\right),$$

and, for any real number a, we have $\phi_i\left(av\right) = a\phi_i\left(v\right)$.

Shapley values have recently gained popularity in XAI as a model-agnostic framework for rationalizing machine learning decisions. However, the high computational demand of extensively enumerating all possible feature coalitions renders the exact computation of Shapley values unfeasible for high-dimensional feature spaces, typical of biology and chemistry. This led to the development of approximation methodologies, from Monte Carlo sampling approaches [236] to local model approximations [8], as the already mentioned SHAP. Using Monte Carlo, the contribution $\phi_i(v)$ can be approximated as

$$\hat{\phi}_i(v) = \frac{1}{M} \sum_{m=0}^{M-1} \left(v(\boldsymbol{x}^m_{+i}) - v(\boldsymbol{x}^m_{-i})\right),$$

where $\boldsymbol{x}^m_{+i}$ is the input feature vector in which a random number of features is replaced with values from a randomly sampled instance, excluding the value of the assessed feature i, and $\boldsymbol{x}^m_{-i}$ is an identical vector, except for the value of feature

i, which is taken from the sampled instance in this case. This procedure allows us to calculate the marginal contribution of i and obtain an approximated version of the Shapley values, also adjusting the trade-off between computation time and approximation accuracy when needed by tweaking the number of sampling steps M.

Regarding SHAP, it approximates a complex machine model in the feature space vicinity of a test instance with a simpler local model based upon a kernel function. Along with approximation methods, SHAP offers the possibility of exactly computing Shapley values for tree-based models, such as RFs, with its TreeSHAP (or TreeExplainer) variant [238]. SHAP-based methodologies have also been introduced and evaluated in chemoinformatics applications [239, 204]. When introducing SHAP, the authors defined a set of desirable properties that must be satisfied by the additive feature attribution methods, which include SHAP and LIME, thereby unifying those methodologies under a common framework. Those properties are *local accuracy*, *missingness*, and *consistency*. Given the original model to explain $\hat{f}$ and a simplified explanation model g, which is a linear function of binary variables x'_i indicating if a feature i is present or not in the original input $\boldsymbol{x}$, the local accuracy property states:

$$\hat{f}(\boldsymbol{x}) = g(\boldsymbol{x}') = \phi_0 + \sum_{i=1}^{|\mathcal{F}|} \phi_i x'_i, \tag{2.3}$$

where ϕ_0 indicates the model output when all features are missing ($x'_i = 0\ \forall i$). Note that by defining $\phi_0 = E_X(\hat{f}(\boldsymbol{x}))$ and setting all features to be present ($x'_i = 1\ \forall i$), this definition is analogous to the efficiency property of the Shapley values (Equation (2.2)).

According to the missingness properties, a missing feature (with simplified input $x'_i = 0$) has an attribution of 0. If a feature is not present in any coalition, its contribution is null. This property is needed to guarantee that missing features won't get an arbitrary attribution value. Even though the latter would not hurt the local accuracy property (since multiplied by $x'_i = 0$) this property forces missing features to have an attribution of 0 (which is correct since they have no impact on the output value) in order to have a unique solution for Equation (2.3). In practice, this property is relevant only for constant features, with no impact on the model (similarly to the dummy property).

The final property is consistency. It asserts when a model changes such that the

marginal contribution of a feature to the value increases or remains the same, its attribution does not decrease. From this property, the Shapley value properties of symmetry, dummy, and linearity follow [8]. Satisfying those properties, it is possible to adapt additive feature attribution methods, like LIME and DeepLIFT, to let them assign approximated versions of Shapley values as feature attributions (SHAP values).

However, approximating Shapley values is not always enough and fails to capture feature importance in some scenarios, like in molecular activity prediction with SVMs, as mentioned in the introduction. This partly motivated our research for the development of a strategy for exact Shapley value computation (Section 4.3): the aforementioned SVERAD methodology.

The Shapley value concept has recently also been extended to graph neural networks. For example, GraphSVX [240] was introduced as a decomposition method for GNNs that uses Shapley values to determine node and node feature contributions. The technique offers post hoc local explanations using a surrogate linear model that allows the approximation of Shapley values. Another method is SubgraphX [241]. It is the first method to be focused on the research of explanation subgraphs only in terms of connected graphs, evaluating the importance that each of them has on the prediction approximating Shapley values for each node coalition. It exploits a Monte Carlo tree search to look for promising coalitions of connected nodes and selects the one associated with the highest Shapley value as the explanation. Finally, GRAPHSHAP [242] was explicitly developed as a motif-focused explanatory approach for generic graph classification with node awareness [243]. This methodology is based on motif masking and uses an approximation kernel for Shapley values to assess the most influential motifs in the graph. None of the Shapley value-based methods present in the literature account for edge importance, although edges represent the links through which information is spread in graphs. Along with the significance that molecular bonds (edges) hold in molecules, this motivated our research in developing our novel edge-centric XAI methodology for GNNs, the already mentioned EdgeSHAPer, presented in Section 4.1.

Chapter 3

Bioinformatics and Network Medicine

The first part of the explainable biomedical deep learning pipeline (Figure 1.1) falls into the domain of bioinformatics and network medicine. Specifically, this chapter covers blocks 1, 2, and partly 3 of the pipeline. As introduced, the first step consists of identifying novel disease-associated genes. Our main goal was to develop a neural network-based method and augment it with explainable artificial intelligence (XAI) capabilities. However, we had to face the problem of training machine learning models, particularly neural networks, in unconventional positive–unlabeled (PU) settings. For this reason, we first had to develop a strategy that enabled proper model learning, which we later used to train a graph neural network (GNN) for disease gene discovery purposes. Interestingly, this training strategy proved effective even as a standalone gene–disease association (GDA) discovery tool and in network-based drug discovery. Therefore, In Section 3.1, we will introduce NIAPU, our PU learning strategy; in Section 3.2, we will present XGDAG, our explainable GNN approach for gene prioritization; and in Section 3.4 we demonstrate how NIAPU can be successfully applied in the context of drug repurposing. Before this, in Section 3.3, we will analyze the phenomenon of gene–gene interaction detection via neural networks and XAI.

3.1 Network-Informed Adaptive Positive–Unlabeled Learning for Disease Gene Identification

The discovery of GDAs is made difficult by incomplete knowledge of biological and physiological processes. When approaching complex, multi-gene diseases and traits, it is hard to disentangle the contribution of each gene, and computational biological approaches for predicting GDAs [244, 245] can support and address experimental

methods (e.g., genome-wide association—GWAS—or linkage studies, among others) which are expensive and time-consuming. The fuzzy background of yet unknown or truly unassociated genes makes the computational identification of disease genes challenging to carry out with accuracy. In machine learning, this setting translates into the ability to identify new positive instances among a set of positive and unlabeled samples, a task known as PU learning [246, 21]. Because associations may exist but may not have been discovered yet, it is not safe to mark unknown associations as negative. Moreover, PU datasets are usually highly unbalanced. In fact, only a small fraction of the entire set of genes in the interactome are associated with a given disease. Training on unbalanced datasets can negatively impinge on the performance of machine and deep learning models, resulting in the need for specific methods for unbalanced learning [247]. PU learning can be addressed through semi-supervised learning algorithms trained using two approaches. In the first one, the set of unlabeled instances is assumed to be a contaminated set of negative instances, and the contamination is considered during the modeling process by weighting the data points or adding penalties on misclassification [248, 65, 249, 250]. In the specific case of gene discovery, this contamination is given by the possibility of the negative instances of containing not yet discovered positive genes. The second approach, called two-step technique, aims at relabeling the instances and then training a supervised learning algorithm [246, 251, 252]. For example, Yang et al. [251] introduced a multi-class labeling procedure considering five different labels, namely positive (P), likely positive (LP), weakly negative (WN), likely negative (LN), and reliable negative (RN), based on a Markov process with restart [253], widely applied in disease gene identification [254, 255, 256]. Then, a supervised learning algorithm is trained on the relabeled data.

Inspired by previous work, we considered the multiclass labeling approach since it allows identifying a set of originally unlabeled items, namely the LP set, whose features are close to that of the items in P. This translates into identifying a small set of genes more likely to contain true positive instances, hence providing a set of new candidate disease genes for prioritization. Going beyond previous approaches, we propose several significant modifications of the multiclass method regarding the distance matrix defining the Markov process and the selection of the different classes. Some of these modifications were needed in order to apply the method to general PU datasets, while others were proposed to make the process of class formation more rigorous and, at the same time, flexible. The approach considered here, being a two-step technique, is based on the separability and smoothness assumptions [21], which require that the features should be able to distinguish between positive and negative instances and, at the same time, instances with similar features should

be more likely to have the same label. Therefore, as a further contribution, we propose the use of specific network-informed features, one of them introduced for the first time, based on protein–protein interaction (PPI) data, which provide a characterization of the topological relationships of all human genes with respect to the set of disease genes. The use of such measures grants a much more precise classification of genes than other topological measures. In particular, the set of seed (positive) genes is identified very precisely as well as the genes closest and farthest to them, as shown in Section 3.1.4.1. The Network-Informed Adaptive Positive–Unlabeled (NIAPU) framework is therefore formed by two components: the Network Diffusion and Biology-Informed Topological (NeDBIT) features and the Adaptive Positive–Unlabeled (APU) labeling algorithm. A visual overview of the workflow can be grasped in Figure 3.1.

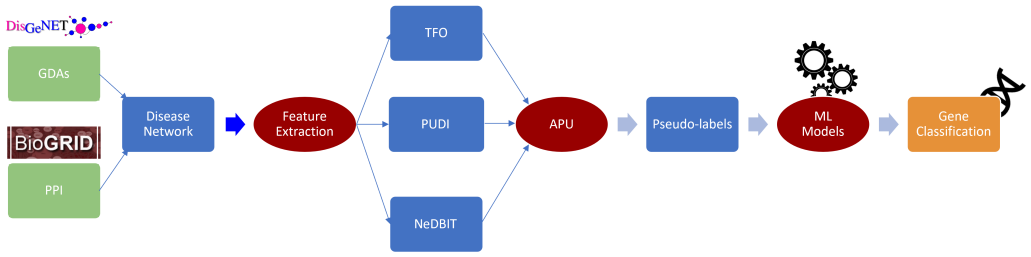


Figure 3.1. The complete NIAPU pipeline. PPI and GDAs are used to obtain a disease-related network. Features are extracted (Section 3.1.3), and APU is applied (Section 3.1.2) to assign pseudo-labels to train machine learning (ML) models for the final gene classification. The assigned pseudo-labels can be used for disease gene discovery purposes (Section 3.1.4.3).

3.1.1 Data Sources and Preprocessing

The proposed methodology exploits two types of data, i.e., reliable PPIs and known GDA data. PPI data provide valuable biological knowledge for the identification of undiscovered disease genes [245, 257, 258, 259, 260]. Human PPI data, i.e., the human interactome, were gathered from the BioGRID dataset (version 4.4.206) [261, 13]. The human interactome is obtained by choosing *Homo sapiens* genes (organism ID 9606), from which we extract a connected network consisting of 19,761 genes and 678,932 nonredundant, undirected interactions. GDAs were derived from DisGeNET (version 7.0) [262, 48, 14], the already mentioned discovery platform containing one of the largest publicly available collections of genes and variants associated with human diseases together with a score denoting the association confidence and significance. Ten diseases were considered: malignant neoplasm of breast (disease ID C0006142,

1074 genes), schizophrenia (C0036341, 883 genes), liver cirrhosis (C0023893, 774 genes), colorectal carcinoma (C0009402, 702 genes), malignant neoplasm of prostate (C0376358, 616 genes), bipolar disorder (C0005586, 477 genes), intellectual disability (C3714756, 447 genes), drug-induced liver disease (C0860207, 404 genes), depressive disorder (C0011581, 289 genes), and chronic alcoholic intoxication (C0001973, 268 genes). The selection criterion for these diseases was the highest cardinality of GDAs in the DisGeNET *curated* dataset to ensure sufficient information for the classification task, especially for neural network models. This dataset version contains GDAs from reliable sources [263, 264, 265, 266, 267, 268]. Instead, to validate the gene discovery results, we relied on the *all genes* DisGeNET dataset, which we refer to as *extended* dataset. The latter contains associated genes from additional sources not present in the curated version [269, 270, 271, 272], and forms a solid base to evaluate the discovery efficacy of computational methods. We checked the presence of each gene from DisGeNET in the BioGRID dataset and we removed the absent ones. After data cleaning, we ended up having a set of associated genes for each disease, denoted by Σ, with their association score $\mathbf{S}$. The latter corresponds to the DisGeNET GDA score, a value ranging from 0 to 1 computed using the number and type of sources (level of curation and model organisms) and the number of publications supporting the association [262]. In particular, we have 1025 genes for disease C0006142, 832 for C0036341, 747 for C0023893, 672 for C0009402, 606 for C0376358, 451 for C0005586, 431 for C3714756, 320 for C0860207, 279 for C0011581, and 255 for C0001973.

3.1.2 Adaptive Positive–Unlabeled Labeling Algorithm

The APU labeling algorithm consists of a multiclass labeling procedure that relies on the labels introduced in previous work [251]: P, LP, WN, LN, and RN. P instances are the known disease genes, RN instances represent the genes whose features are the furthest from the average features in the P set, while the remaining labels are assigned through a Markov process with restart [253]. The novelty of the proposed method is the construction of a new transition matrix starting from the distance matrix between the features of the genes. The matrix needs to be normalized in order to preserve the total transition probability of the state vector whose initial value is different from zero only for genes in the P and RN classes. Moreover, the class selection was made flexible using an adaptable quantile separation instead of fixed thresholds. These characteristics were implemented to make the process of class formation more rigorous and, at the same time, more flexible, hence easily adaptable to different settings, datasets, and needs.

Let $\mathcal{V}$ be a set whose generic i^{th} element $v_{i=1,\dots,n}$ is characterized by the couple

$(\boldsymbol{x}_i, y_i)$ where $\boldsymbol{x}_i \in [0,1]^d$ represent the feature vector, and $y_i \in \{0,1\}$ the initial label. The APU algorithm is defined by the following steps.

Step 1: Compute the matrix $\boldsymbol{W}$, whose elements w_{ij} are defined as follows:

$$w_{ij} = \begin{cases} 1 - \frac{e_{ij}-m}{M-m} & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases},$$

where $e_{ij} = \sum_k \left(x_i^k - x_j^k\right)^2$, $m = \min_{ij}\{e_{ij}\}$, and $M = \max_{ij}\{e_{ij}\}$. The symmetric matrix $\boldsymbol{W}$ contains the similarity score between elements i and j.

Step 2: Compute the reduced matrix $\boldsymbol{W}_r$ as follows:

$$w_{r,ij} = \begin{cases} w_{ij} & \text{if} \quad w_{ij} > q_{\boldsymbol{w}} \\ 0 & \text{otherwise} \end{cases}.$$

The threshold $q_{\boldsymbol{w}}$ is computed as a given quantile of the distribution of the elements in the matrix $\boldsymbol{W}$ in order to exclude from the propagation process links between poorly related elements. To obtain a proper Markov process, i.e., preserving the probability distribution, the matrix $\boldsymbol{W}_r$ must be normalized as $\boldsymbol{W}_n = \boldsymbol{D}^{-1}\boldsymbol{W}_r$, where $\boldsymbol{D}$ is the diagonal matrix with elements $d_{ii} = \sum_j w_{r,ij}$.

Step 3: Initialize the propagation process with the initial state vector $\boldsymbol{g}_0$ defined as follows. Let $|P|$ be the cardinality of P (set of seed genes) and $\hat{\boldsymbol{x}} = \left(\hat{x}^1, \ldots, \hat{x}^d\right)$, where $\hat{x}^k = 1/|P| \sum_{i \in P} x_i^k$, be the average features of P. The RN genes are chosen to be the ones having the most distant features from $\hat{\boldsymbol{x}}$. $|P|$ most distant genes from $\hat{\boldsymbol{x}}$ can be selected in order to keep the classes balanced. Then, the i-th element of $\boldsymbol{g}_0$ is defined as

$$g_{0,i} = \begin{cases} 1 & \text{if} \quad i \in P \\ -1 & \text{if} \quad i \in RN \\ 0 & \text{otherwise} \end{cases}.$$

When needed, a different number of RN genes can be selected. In this case, the initial value of the RN genes in the state vector $\boldsymbol{g}_0$ must be set to $-|P|/|RN|$ so that the two distributions of positive and negative values are balanced in $\boldsymbol{g}_0$, with the sum of its elements equal to zero.

Step 4: Define a Markov process with restart as

$$\boldsymbol{g}_r = (1-\alpha)\,\boldsymbol{W}_n^t \boldsymbol{g}_{r-1} + \alpha \boldsymbol{g}_0, \tag{3.1}$$

where the parameter α is usually set to 0.8 [251, 256]. Starting from the state vector $\boldsymbol{g}_0$ the dynamics in Equation (3.1) converges in the stationary state $\boldsymbol{g}_\infty$, numerically reached when $|\boldsymbol{g}_r - \boldsymbol{g}_{r-1}| < 10^{-6}$.

Step 5: Use $\boldsymbol{g}_\infty$ to assign the remaining labels. Selecting only the elements that belong neither to P nor to RN, the values of the asymptotic distribution of those elements are sorted, and the ranking of the corresponding elements is used to form the remaining classes: LP, WN, and LN. A simple rule is to divide the ranking into three equal parts and identify LP samples with the first third, WN with the second third, and LN with the third third. However, depending on the type of analysis and the problem addressed, any ranking division can be considered acceptable.

Step 6: Classification. A machine learning classifier is trained over the dataset with the new propagated labels. To test the efficacy of the method, three different algorithms have been used: two classic approaches, namely random forest (RF) [107] and support vector machine (SVM) [6, 7], and a neural network model, the multilayer perceptron (MLP) [131].

3.1.3 NeDBIT Features

The APU labeling procedure can be used with any set of features. However, we devised disease-specific features that adequately characterize the set of positive genes and distinguish them from the rest of the classes. The NeDBIT features include two network diffusion-based features, heat diffusion and balanced diffusion, and two biology-informed topological metrics, NetShort and NetRing. Network diffusion methods are widely used in disease gene discovery processes [273, 274, 275]. We coupled network diffusion methods and innovative topological-based features to make the most out of the combined predictive power of both approaches. Moreover, all the

features are computed exploiting the association score **S**. In this way, the NeDBIT features, not assigning the same weight to all seed genes, are more significant for the disease under investigation.

3.1.3.1 Heat Diffusion Feature

This feature is obtained using a heat diffusion process over the network [276]. Starting with a distribution of weights, with positive values only on the seed genes, their evolution is determined by using the diffusion equation on graph [277]

$$\mathbf{z}'(t) + \mathbf{L}\mathbf{z}(t) = 0, \tag{3.2}$$

where $\mathbf{L}$ is the Graph Laplacian matrix, $\mathbf{L} = \mathbf{K} - \mathbf{A}$, $\mathbf{K}$ is the diagonal matrix with the degree of nodes on the diagonal, namely $\mathbf{K}_{ii} = k_i$, and $\mathbf{A}$ is the adjacency matrix of the PPI. The weights at time t are given by the formal solution of Equation (3.2):

$$\mathbf{z}(t) = \exp\left(-\mathbf{L}t\right)\mathbf{z}(0), \tag{3.3}$$

where exp is the exponential of the matrix. Regarding the initial distribution of weights, we assign $z_i(0) = s_i$ for seed genes in Σ and 0 otherwise, where s_i is the GDA score.

3.1.3.2 Balanced Diffusion Feature

This feature is obtained by using the diffusion in Equation (3.2) but with a different version for the Graph Laplacian matrix, i.e., $\mathbf{L}_b = \mathbf{I} - \mathbf{K}^{-1}\mathbf{A}$. The weights at time t are obtained as in Equation (3.3) by using operator $\mathbf{L}_b$, and the initial weights are given as for the previous measure.

This form of the graph diffusion operator differs from the heat diffusion in the fact that the operator $\mathbf{L}$ diffuses the same amount of score for each link, whereas $\mathbf{L}_b$ diffuses the same amount of score for each node. This implies a different short-time behavior of the diffusion process on the graph. In fact, in heat diffusion, well-connected nodes are drained faster, while in balanced diffusion, all the nodes diffuse information at the same pace.

3.1.3.3 NetShort

The NetShort measure [278] is based on the idea that a generic node is topologically important for a disease if a large number of seed nodes must be traversed to reach it. For each node, the weights are assigned as follows:

$$w_{ij} = a_{ij}\frac{2}{\tilde{s}_i + \tilde{s}_j}, \quad \text{where} \quad \tilde{s}_i = \begin{cases} \frac{s_i}{\max \mathbf{S}} & \text{if} \quad i \in \Sigma \\ \alpha\frac{\min \mathbf{S}}{\max \mathbf{S}} & \text{if} \quad i \notin \Sigma \end{cases},$$

where $\min \mathbf{S}$ and $\max \mathbf{S}$ are the minimum and the maximum association scores, α is the penalization parameter given to non-seed nodes, and a_{ij} is the element in position (i, j) in the adjacency matrix $\mathbf{A}$. We use $\alpha = 0.5$ so that all non-seed nodes have normalized score $\tilde{s}_i = \frac{1}{2}\frac{\min \mathbf{S}}{\max \mathbf{S}}$ while seed nodes have normalized score $\frac{\min \mathbf{S}}{\max \mathbf{S}} \leq \tilde{s}_i \leq 1$. Then, the NetShort measure NS_i for node i is defined as

$$NS_i = \sum_{j \neq i} \frac{1}{d_{ij}},$$

where d_{ij} is the length of the weighted shortest path from i to j, computed using the assigned weights. In this way, links connecting seed genes are favored and links connecting non-seed genes are penalized.

3.1.3.4 NetRing

The NetRing measure, introduced for the first time with our methodology, is based on the concept of ring structure [279] generalized to a set of seed nodes. Starting from seed nodes, a partition of the graph in subgraphs, or rings, is introduced with the following property:

$$R(l) \equiv \left\{ j \in \mathcal{V} \;\middle|\; \min_{i \in \Sigma} l_{ij} = l \right\},$$

where l_{ij} is the (unweighted) shortest path length from i to j. $R(l)$ contains all the non-seed nodes with a minimal distance l from, at least, one seed node. From the definition follows that $R(0) \equiv \Sigma$ (set of seed nodes), $R(l_1) \cap R(l_2) = \emptyset$ if $l_1 \neq l_2$ and $\mathcal{V} = \bigcup_{l=0}^{L} R(l)$, where L is the highest value of the minimal distance from non-seed

nodes to seed nodes.

An initial rank defined by means of the association score $\mathbf{S}$ is computed as

$$\hat{r}_i = \begin{cases} 1 - \frac{s_i}{\max \mathbf{S}} & \text{if } i \in \Sigma \\ 1 & \text{otherwise} \end{cases},$$

then the NetRing measure r_i of node i is defined as

$$r_i = \begin{cases} \alpha \hat{r}_i + (1-\alpha) \dfrac{1}{k_i} \displaystyle\sum_{j|A_{ij} \neq 0} \hat{r}_j & \text{if } i \in \Sigma \\ l_i + \dfrac{1}{k_i} \left(\displaystyle\sum_{j \in O_i} \hat{r}_j + \sum_{j \in R_i(l_i-1)} r_j - (l_i - 1) \right) & \text{otherwise} \end{cases},$$

where the score for seed genes is a convex combination of the initial rank $\hat{r}_i$ and the average of the initial rank of the neighbors of the node so that seed nodes having many seed nodes as neighbors have a higher rank. The rank of non-seed nodes is obtained by summing the level of the ring and the average of two terms, i.e., the number of genes belonging to the same or higher rings ($O_i = \{j \notin R(l_i - 1) | A_{ij} \neq 0\}$) and the sum of the rank of genes in the lower ring ($R_i(l_i - 1) = \{j \in R(l_i - 1) | A_{ij} \neq 0\}$) corrected by the ring level. The correction is introduced to make the rank r_j comparable with $\hat{r}_j$. This measure rewards non-seed nodes close to high-ranking nodes in the previous ring and linked with a few nodes of the same or higher rings.

The ring concept leads to the introduction of a ranking between nodes, i.e., ring zero includes all seed nodes, ring one includes all the nodes that are directly connected to at least one seed node, and so on, as the ring level grows. But, when dealing with seed nodes representing disease genes, it is also evident that not all the nodes in each ring are equivalent. For example, there may be a node in the first ring that has only one direct contact with a seed node, while another node in the first ring may be directly connected to many seed nodes (it is to be noted that this concept resembles and extends connectivity significance [50]). So, to rank nodes belonging to the same ring, it is essential to consider the number of nodes on the lower ring a node is connected to, together with their ranks.

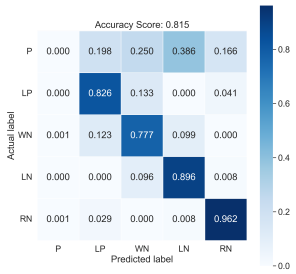
3.1.4 Results and Performance Analysis

The performance of NIAPU is tested on the ten disease datasets detailed in Section 3.1.1. Section 3.1.4.1 is devoted to testing the performance of NIAPU (APU + NeDBIT) against the implementation of the APU labeling algorithm with two different sets of features commonly used when dealing with disease gene identification. Section 3.1.4.2 analyzes the performance of NIAPU in identifying candidate disease genes. To this end, a subset of seed genes is masked out (prior to the computation of the NeDBIT features and the application of the APU algorithm) to see whether such genes are predicted as LP. Section 3.1.4.3 compares NIAPU with other disease gene identification algorithms, while Section 3.1.4.4 presents results from an enrichment analysis of the candidate disease genes obtained by the NIAPU methodology. For our experiments, we set the threshold q_w for the removal of weak links (see Step 2 of Section 3.1.2) to 0.05 and defined the RN set to contain 20% of the genes.

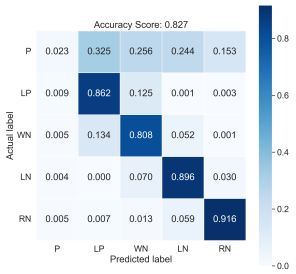
3.1.4.1 Classification with NeDBIT Features

The effectiveness of the NeDBIT features is tested by comparing NIAPU against the implementation of the APU labeling algorithm with two different sets of features: the first (PUDI), computed following previous work [251], is based on topological features [280] and functional information based on the semantic similarity of Gene Ontology (GO) terms [281]. The latter are indicative of gene functions in terms of molecular function, biological process, and cellular component. The second set of features (TFO) includes simple topological, functional, and ontological features like degree, degree centrality, betweenness centrality, eigenvector centrality, clustering coefficient, closeness centrality, current closeness, and GO terms from the biological process domain [282]. The comparison is carried out in terms of out-of-sample classification performance; namely, the ten datasets were split into training set (70%) and test set (30%), keeping class balance. Then, we trained the three machine learning algorithms defined in Step 6 of Section 3.1.2 for the three different applications of the APU algorithm. We report in Figure 3.2 the results for malignant neoplasm of breast.

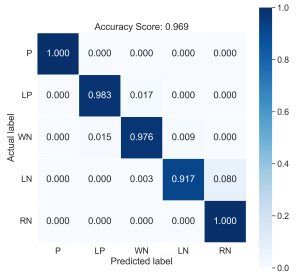
The comparison among TFO, PUDI, and NeDBIT features shows that the latter are far superior. The joint usage of APU and NeDBIT features (NIAPU) successfully discriminates the class P from the rest of the genes and better separates the pseudo-classes LP, WN, LN, and RN. Regarding the pseudo-classes, the identification performances were also satisfying using TFO and PUDI features, even if with a drop in accuracy compared to NeDBIT. This highlights the effectiveness of the APU label assignment. RF and MLP delivered the best performances. Regarding SVM, LN samples were sometimes misclassified as WN or RN.



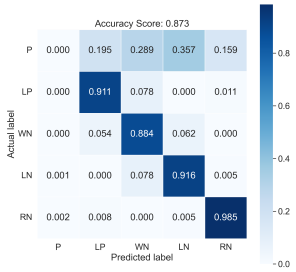
(a) MLP + TFO features



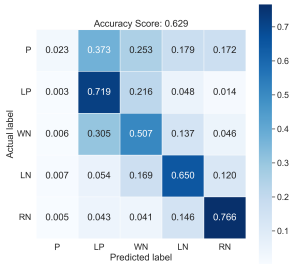
(b) MLP + PUDI features



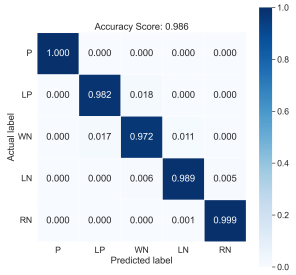
(c) MLP + NeDBIT features



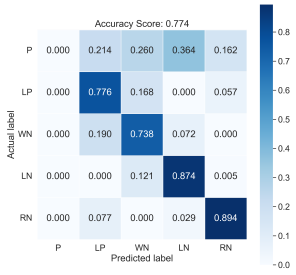
(d) RF + TFO features



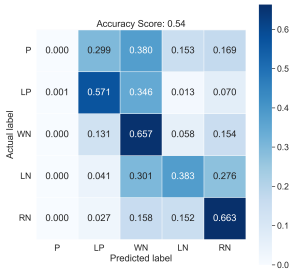
(e) RF + PUDI features



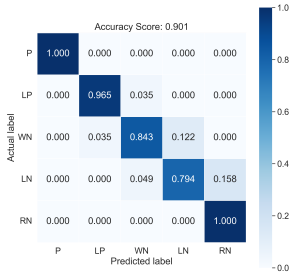
(f) RF + NeDBIT features



(g) SVM + TFO features



(h) SVM + PUDI features



(i) SVM + NeDBIT features

Figure 3.2. Confusion matrices for multiclass classification on malignant neoplasm of breast (C0006142). The APU labeling and the newly defined NeDBIT features allow for a better and clearer distinction of the P class and the pseudo-classes.

Overall, for P and RN classes, NIAPU classification is almost perfect since the NeDBIT features allow those classes to be coherently separated from the others since they grasp the topological aspects of the set of seed genes as a whole, assigning lower and lower weights to genes that are progressively "far" from the set of seed genes. The usage of TFO and PUDI features fails in the identification of the class P. This is due to the fact that such sets of features are too general and cannot capture the

specificity of a given disease. For the rest of the classes, the performances are good, but some genes are misclassified. This is due to the label assignment via quantiles, which obviously introduces some arbitrary noise at the boundaries.

3.1.4.2 Performances in Disease Gene Identification

We tested the ability of NIAPU to identify new candidate genes. We performed a validation by excluding 20% of seed genes, setting them as unlabeled in the computation of the NeDBIT features and in the APU labeling algorithm. We repeated the procedure five times with non-overlapping gene sets. We investigated whether NIAPU was able to classify the removed positive genes as LP. The results for malignant neoplasm of breast are reported in Table 3.1.

Table 3.1. Labeling of the unlabeled seed genes by NIAPU for malignant neoplasm of breast (C0006142). Results are intended as average with standard deviation over the five runs (GDAS: association score **S**).

Label	Genes (%)	Genes (number)	GDAS mean	GDAS median	GDAS mode
LP	45.659 ± 1.362	93.600 ± 2.793	0.383 ± 0.016	0.346 ± 0.019	0.320 ± 0.045
WN	27.415 ± 0.636	56.200 ± 1.304	0.343 ± 0.013	0.318 ± 0.011	0.300 ± 0.000
LN	17.659 ± 4.436	36.200 ± 9.094	0.324 ± 0.012	0.303 ± 0.004	0.300 ± 0.000
RN	9.268 ± 3.650	19.000 ± 7.483	0.322 ± 0.013	0.303 ± 0.004	0.300 ± 0.000

On average, around 46% of unlabeled seed genes fell in the LP class, while the rest fell in a decreasing classification trend toward the RN class. We also observed a clear correspondence between the labeling and the association score: the higher the score, the more likely the gene will be found in the LP class. This underlines the influence of scores on the NeDBIT features. Analogous results were obtained for the rest of the diseases. Furthermore, aggregated results related to machine learning classification for all the diseases are reported in Table 3.2. RF and MLP identified all the classes with high scores, while SVM reported lower metrics, particularly for the LN class. Therefore, NIAPU proved robust also in more challenging settings with reduced seed gene sets.

3.1.4.3 Comparison with Gene Prioritization Tools

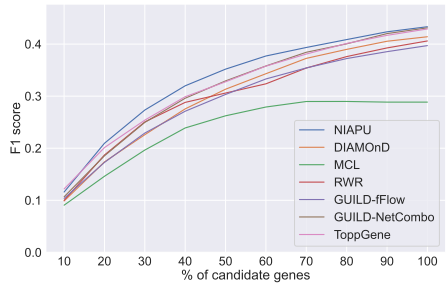
We compared the predictive performance of NIAPU in the identification of candidate disease genes against known gene discovery algorithms, already introduced in Section 2.1, namely DIAMOnD [50], Markov clustering (MCL) [283, 54], random walk with restart (RWR) [254, 56], ToppGene [60], and two variants of GUILD [58], one exploiting the NetCombo measure and the other based on Functional Flow

Table 3.2. Classification scores as pooled mean and standard deviation (over all the diseases). Five runs were performed for each disease, masking out 20% of seed genes.

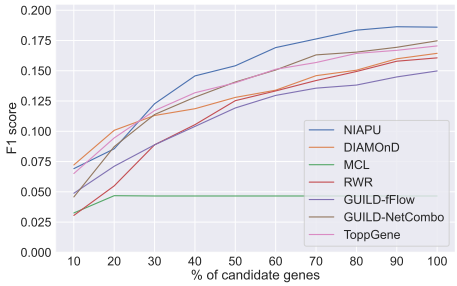
Label	Precision	Recall	F1 score
MLP			
P	0.994 ± 0.011	0.998 ± 0.007	0.996 ± 0.007
LP	0.972 ± 0.013	0.972 ± 0.016	0.972 ± 0.012
WN	0.955 ± 0.020	0.915 ± 0.022	0.933 ± 0.019
LN	0.835 ± 0.021	0.744 ± 0.042	0.782 ± 0.019
RN	0.731 ± 0.037	0.860 ± 0.036	0.788 ± 0.024
Macro avg	0.898 ± 0.008	0.898 ± 0.007	0.894 ± 0.008
Weighted avg	0.884 ± 0.009	0.876 ± 0.009	0.876 ± 0.009
Accuracy	0.876 ± 0.009		
RF			
P	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
LP	0.984 ± 0.005	0.984 ± 0.005	0.984 ± 0.005
WN	0.977 ± 0.007	0.976 ± 0.007	0.977 ± 0.006
LN	0.982 ± 0.005	0.986 ± 0.004	0.984 ± 0.004
RN	0.991 ± 0.003	0.987 ± 0.004	0.989 ± 0.003
Macro avg	0.987 ± 0.003	0.987 ± 0.003	0.987 ± 0.003
Weighted avg	0.984 ± 0.004	0.984 ± 0.004	0.984 ± 0.004
Accuracy	0.984 ± 0.004		
SVM			
P	0.998 ± 0.004	1.000 ± 0.000	0.999 ± 0.002
LP	0.845 ± 0.043	0.719 ± 0.071	0.767 ± 0.032
WN	0.635 ± 0.135	0.726 ± 0.108	0.625 ± 0.102
LN	0.625 ± 0.191	0.559 ± 0.026	0.419 ± 0.025
RN	0.366 ± 0.224	0.500 ± 0.004	0.38 ± 0.011
Macro avg	0.694 ± 0.066	0.701 ± 0.013	0.638 ± 0.022
Weighted avg	0.641 ± 0.077	0.642 ± 0.017	0.568 ± 0.029
Accuracy	0.642 ± 0.017		

(fFlow) [284]. For this analysis, we relied on the extended GDA dataset provided by DisGeNET. We assigned the labels using NIAPU on the curated version of the dataset. Then, we investigated whether the seed genes contained in the extended version (but not in the curated one) fell into the LP class. We considered the ranking retrieved by NIAPU at different quantile thresholds. In Figure 3.3, we report the results of this comparison in terms of F1 score.

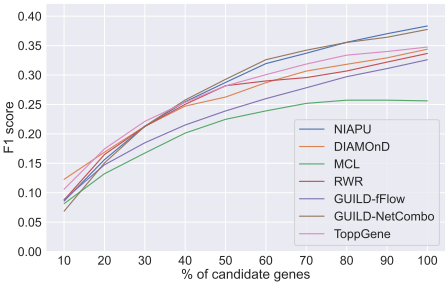
Most of the time, our methodology outperformed or was at par with the state-of-the-art algorithms, being often the best-performing method when looking for a large



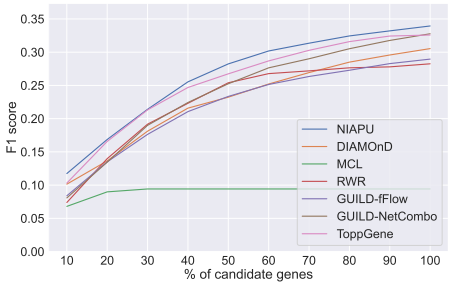
(a) Malignant neoplasm of breast (C0006142)



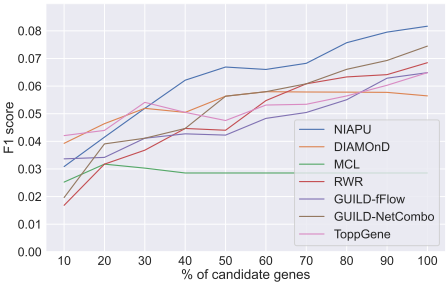
(b) Schizophrenia (C0036341)



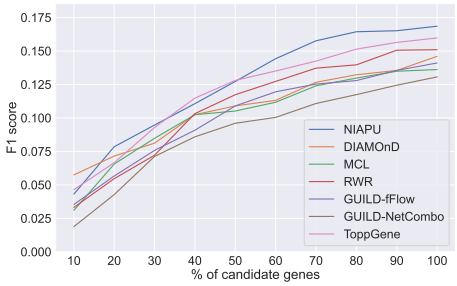
(c) Colorectal carcinoma (C0009402)



(d) Malignant neoplasm of prostate (C0376358)



(e) Bipolar disorder (C0005586)



(f) Intellectual disability (C3714756)

Figure 3.3. Gene discovery performances in terms of F1 score. Results are reported for six representative diseases at increasing numbers of candidate genes considered as a percentage of the total number of associated genes in the extended dataset, which is different for each disease.

number of candidate genes and of comparable performances for lower ones. Indeed, DIAMOnD performs at its best when considering a low ratio (10-20%) of predicted genes. In contrast, NIAPU performs well for low and high percentages of candidate genes, outperforming DIAMOnD in the latter case. In fact, as stated by the authors themselves, DIAMOnD becomes unreliable when exceeding 200 predicted genes [50].

3.1.4.4 Enrichment Analysis

To further evaluate our results, we performed an enrichment analysis of the first 100 predicted genes in the LP class from the validation on the extended GDA dataset. Enrichment analysis is a process used to identify genes that are over-represented in a given disease network, biological pathway, or gene ontology, and that can have significant associations with specific phenotypes. This can bring further interesting insights that can validate methodologies and experiments or lead to new discoveries. This is useful to understand if the genes retrieved by our methodology are involved in biological pathways, gene ontologies, or other disorders related to the investigated diseases. This analysis was performed using the Enrichr online tool [285, 286, 287].

Notably, the LP genes selected for enrichment do not correspond to any of the curated GDA disease genes; therefore, among the enriched diseases, we cannot expect to find the same disease for which the gene discovery process is carried out. Instead, among the enriched terms (diseases, GO terms, or pathways), we should be able to find diseases and biological processes that are related to the disease under scrutiny. We report the enrichment analysis results in Table 3.3. In particular, we present the top enriched diseases or biological processes for each analyzed disease, together with literature references that endorse such relevant links. The fact that there is evidence in the literature of relationships and shared biological mechanisms between the analyzed diseases and enriched terms is additional proof of the validity and efficacy of the NIAPU disease gene discovery process.

3.1.5 Observations

In this section, we presented the first contribution of this thesis, the NIAPU algorithm, which fits the typical problem of the computational identification of previously unknown disease genes in the context of positive–unlabeled learning. The advantage of the proposed method is that it allows accurate characterization of the positive samples (P set)—via the NeDBIT features—and refined control of the likely positive samples (LP set)—via the APU labeling procedure—which, extracted from the set of unlabeled elements, contains, with the highest probability, elements related to the disease of interest. Moreover, NIAPU turned out to be an effective labeling procedure, allowing machine and deep learning models to be trained appropriately and deliver highly accurate classification performances. As for disease gene identification, NIAPU proved efficient in two experiments. In the first one, masking out a subset of seed genes, it turned out that ~46% of those fell in the LP class. In the second one, assigning labels using NIAPU on the curated version of the DisGeNET dataset and then searching for the seed genes of the extended version only, the predictive

Table 3.3. Enrichment analysis of the LP genes predicted for the ten diseases of interest. The top enriched diseases, GO terms, and pathways are reported, along with notes about disease relationships and main reference articles.

Disease	Enriched disease/GO/pathway	Relationships and references
C0036341 Schizophrenia	KEGG GO:0042981 Regulation of apoptotic processes	Apoptotic engulfment pathway involved in schizophrenia (increased risk) [288].
C0005586 Bipolar disorder (BD)	KEGG GO:0042981 Regulation of apoptotic processes	Observed relationship between mitochondrial dynamics and dysfunction and the apoptotic pathway activation and the pathophysiology of BD [289].
C0006142 Malignant neoplasm of breast	Leukaemia	Therapy-related myeloid neoplasms may be part of a cancer-risk syndrome involving breast cancer [290].
C0009402 Colorectal carcinoma (CRC)	Ovarian cancer (OC)	GCNT3 might constitute a prognostic factor also in OC and emerges as an essential glycosylation-related molecule in CRC and OC progression [291].
C0011581 Depressive disorder	Parkinson	Neurobiological investigations suggest that depression in Parkinson's disease may be mediated by dysfunction in mesocortical/prefrontal reward, motivational, and stress-response systems [292].
	GO:0043066 Negative regulation of apoptotic processes	Evidence of local inflammatory, apoptotic, and oxidative stress in major depressive disorder [293].
C0023893 Liver cirrhosis	Parkinson	Parkinson's disease among the neurological complications in advanced liver cirrhosis mediated by manganese [294].
C0376358 Malignant neoplasm of prostate	Melanoma	Diagnoses of cutaneous melanoma may be associated with prostate cancer incidence [295].
C3714756 Intellectual disability	Dementia	People with intellectual disability are at higher risk of dementia than the general population [296].
C0860207 Chronic alcoholic intoxication	Ovarian cancer (OC)	Alcohol consumption might be associated with the risk of OC in specific populations or in studies with specific characteristics [297].
	KEGG Estrogen signaling pathway	Association of increased estrogen level and increased alcohol use in females [298].
C0001973 Drug-induced liver disease	Leigh Syndrome (LS)	Valproate, listed as a cause of drug-induced acute liver failure, can cause mitochondrial dysfunction and should be avoided in LS patients [299].

performance of the method outperformed or was at par with the state-of-the-art algorithms for disease gene discovery.

Methodologies like NIAPU are meant to prioritize genes for subsequent experimental validation. Computational prioritization methods aim at reducing the pool of genes to test in clinical or laboratory studies by providing a ranking of candidate genes with a higher likelihood of association with the disease.

It is worth noting that the NeDBIT features are designed to be able to use link-weighted and node-weighted graphs and that, by having increasingly accurate PPIs, we expect increasingly good results from the application of NIAPU. On the other hand, the NIAPU methodology is clearly influenced by the reliability of seed genes, the association score assigned to them, and the background network topology (here, the PPI network and its reliability). Indeed, GDA datasets may be affected by disease–gene association bias due to the quantity of research on a given disease or trait. In this regard, a recent systematic review [300] demonstrated that 87.7% of all genes could be associated with cancer. This indicates that given the massive amount of research focused on cancer, which also applies to other types of diseases, the definition "associated with" is to be checked carefully and critically. The use of datasets that are as error-free, unbiased, and reliable as possible (e.g., using an interactome validated in the specific pathological context, as we will see in Section 3.4, possibly with weighted PPIs) could improve the classification performance of the method [301].

NIAPU delivered accurate and promising results, and its label propagation procedure allowed proper training of machine learning models in PU settings, particularly for gene prioritization purposes. For these reasons, as the next section will show, we relied on NIAPU as the foundation of XGDAG, our explainable graph neural network strategy for disease gene discovery. NIAPU was published in *Bioinformatics* by Oxford University Press [22].

3.2 Explainable Gene–Disease Association via Graph Neural Networks

The work just presented showed that framing gene discovery as a PU learning problem is strategic. The effectiveness of labeling propagation methods allows for proper training of machine and deep learning models, as we demonstrated with NIAPU. For those reasons, we relied on the NIAPU capabilities for the definition of the node features and the label propagation system to serve as the base for our explainable deep learning model. Given the network-like nature of biological datasets,

such as PPIs, the methodology presented in this section for gene prioritization is based on GNNs. As we explained in Chapter 2, those models are able to leverage graph-structured data and capture the information flowing throughout the network. However, we need to enable proper learning for the GNN in the PU setting of gene discovery: we use NIAPU for this purpose. After applying NIAPU, we trained a GraphSAGE [302] model over the propagated labels using the NeDBIT features. Then, an explanation phase generates the *explanation subgraph* for the associated genes that we use to expand the set of candidate genes for further analysis. We make the hypothesis that this set may contain newly associated genes, following the connectivity significance principle [50], according to which a seed gene is likely to be connected to other seed genes. At first, we explore different XAI methods to determine the top-performing ones, and then we compare those selected with several state-of-the-art methods for disease gene identification. We call our proposed method XGDAG (eXplainable Gene–Disease Associations via Graph neural networks).

To the best of our knowledge, XGDAG is the first method to use an XAI-based solution in the context of PU learning for disease gene prioritization with GNNs. Its main novelty lies in the innovative use of the explainability results. Commonly, XAI is used as a passive tool to support and rationalize model decisions. In our case, explainability tools have an active role in the computation of the final ranking, given that the new candidate genes are directly extracted from the explanation subgraphs (see Section 3.2.1.3). This approach drastically diverges from previous attempts to use XAI for GNNs for a similar task. Indeed, previous work [303] proposed the use of XAI to weight patient-specific PPIs before applying clustering for disease module detection. Even in this case, the use of XAI can be regarded as a support tool to enhance the output of other methods rather than an active tool to produce the final results.

The PPI, the GDA data sources, and the diseases considered in this study are the same used in Section 3.1 to allow for a coherent comparison with NIAPU and other strategies.

3.2.1 Methodology

As aforementioned, we frame gene discovery as a PU learning problem. Our method is a three-step procedure that consists of (i) applying the NIAPU label propagation methodology to assign pseudo-labels to enable proper PU learning, (ii) training a GNN GraphSAGE model, and (iii) using explainability strategies for GNNs to compute explanation subgraphs for gene prioritization and define new putative

disease genes. We now explain these steps, depicted in Figure 3.4.

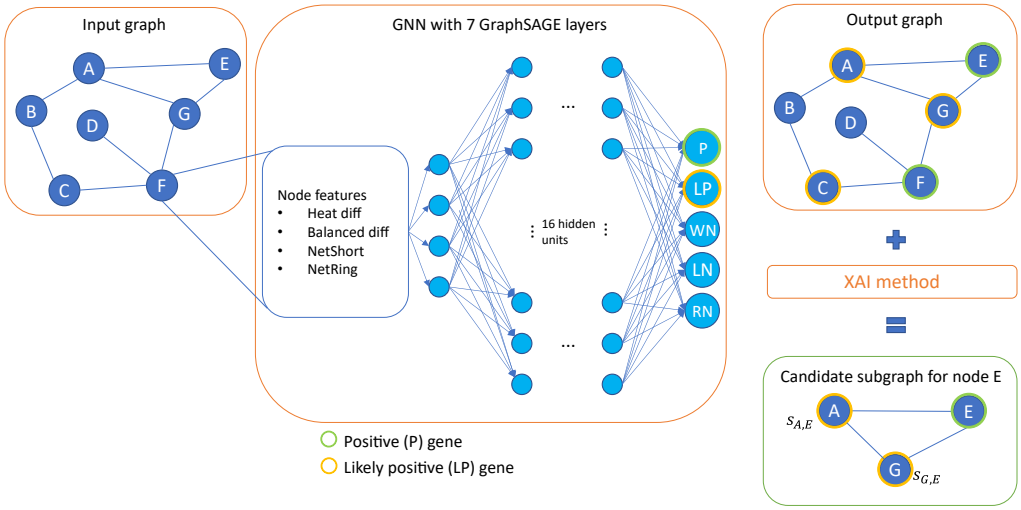


Figure 3.4. The XGDAG framework. A graph based on a PPI network enriched with GDA information, node features, and pseudo-labels is fed into a graph neural network. After the network has been trained, the predictions for the positive (P) genes are explained using an XAI methodology. Next, the nodes that appear in both the explanation subgraph and in the likely positive (LP) set are marked as candidate genes for prioritization.

3.2.1.1 Label Propagation

Our dataset is a PU dataset, in which a gene can be associated with a disease (positive) or not (unlabeled). Given the already stressed problems of working with PU data, label propagation procedures are used to assign pseudo-labels to unlabeled instances, with a two-fold benefit: avoid the bias introduced by setting the unlabeled instances as negative and obtain a more balanced dataset.

Using NIAPU, we assign pseudo-labels to unlabeled genes according to the likelihood of association: likely positive (LP), weakly negative (WN), likely negative (LN), and reliably negative (RN), as in Section 3.1. We make use of the whole pipeline as described in Section 3.1.2, using the NeDBIT features (Section 3.1.3). We remind that the NeDBIT features used in NIAPU are computed taking into account the seed genes (represented by the class P). For this reason, for each disease, we have a different set of features assigned to the genes able to properly characterize the disease itself.

3.2.1.2 Graph Neural Network Model and Training

After the label propagation, we obtain a dataset in which previously unlabeled items are labeled with the most suitable pseudo-label. As Step 6 of the NIAPU pipeline, we train a GraphSAGE [302] model. It works with an inductive learning procedure that learns the embedding of a node, assuming that the nodes in the same neighborhood have similar features. It does that by learning aggregator functions that generate node embeddings relying upon a node's features and neighbors. A GraphSAGE layer, as defined in the PyTorch Geometric implementation we used [304], that generates the embedding $\mathbf{x}'_i$ for node i, after the application of a nonlinear activation function σ, has the following formula:

$$\mathbf{x}'_i = \sigma\left(\mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2 \cdot \text{mean}_{j\in\mathcal{N}_{(i)}}\mathbf{x}_j\right), \tag{3.4}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weights learned by the neural network, $\mathbf{x}_i$ is the feature vector for node i, $\mathcal{N}_{(i)}$ is the 1-hop neighborhood of node i, and $\mathbf{x}_j$ is the feature vector for the neighbor node j. The mean function aggregates information from all the neighboring nodes without applying any sampling. In our case, σ is a ReLU function [305]. The use of this GNN is also suitable for dynamic graphs, as it is able to generate embeddings of new nodes without the need to retrain the model; only node features and neighbor node information are needed. Because a single layer aggregates information at a distance of 1 hop and the diameter of our network is 7, we employ a 7-layer GraphSAGE GNN to gather the information flowing through the whole network. Working with deep GNNs may cause oversmoothing [306], which consists in the degradation of the model's performance as the number of layers increases. To guarantee that this does not occur in our case, we tested different architectures with different depths, obtaining the best performance with 7 GraphSAGE layers via a competitive study. We trained the model using Adam optimizer [307] with learning rate set to $1e-3$ and weight decay to $5e-4$ for a maximum of 40,000 epochs, employing an early stopping procedure when the loss reaches a plateau. To train and evaluate the model, we split the dataset into training (70%), validation (15%), and test sets (15%), maintaining the balance of the classes between the sets. The performances of the GNN on the test set are summarized in Table 3.4.

3.2.1.3 Explainability Phase

The next step, after the training of the model, is to explain its predictions. For that, we have tested several XAI techniques on top of XGDAG. These methods

Table 3.4. Average results with standard deviation over the ten diseases for the GNN model.

Label	Precision	Recall	F1 score
P	0.956 ± 0.033	0.962 ± 0.064	0.958 ± 0.040
LP	0.876 ± 0.082	0.911 ± 0.077	0.888 ± 0.046
WN	0.861 ± 0.068	0.815 ± 0.110	0.831 ± 0.059
LN	0.868 ± 0.046	0.835 ± 0.066	0.850 ± 0.044
RN	0.858 ± 0.055	0.886 ± 0.060	0.871 ± 0.047
Macro avg	0.884 ± 0.027	0.882 ± 0.026	0.879 ± 0.028
Weighted avg	0.869 ± 0.031	0.863 ± 0.034	0.862 ± 0.035
Accuracy	0.863 ± 0.034		

output a subgraph of the original graph, the explanation subgraph, which contains the most influential nodes for the prediction. Our method applies one explainability technique to the positive genes P. For each explained node n, we thus obtain the explanation subgraph G_n. Every node in G_n has an importance score assigned (which depends on the XAI method used). G_n may contain nodes belonging to different pseudo-classes. To enhance the accuracy of the results, we filter G_n by keeping only the genes that the GNN predicted to be LP, which are more likely to be associated genes according to the NIAPU labeling. We thus obtain a reduced explanation subgraph, the candidate subgraph G_n^{LP}. We repeat this process for every node in P. If a node i appears in more candidate subgraphs, it is more likely to be associated with the disease, as per the connectivity significance principle [50]. We take this into account as follows: we keep track of the number M_i of subgraphs in which node i appears and of its cumulative importance score S_i, obtained by summing all the importance scores s_{ij} that node i has in the prediction of each node j—we assume that $s_{ij} = 0$ if i is not in G_j. Every gene i is then assigned a tuple (M_i, S_i). Finally, we obtain a ranking of candidate genes by sorting all the genes in the candidate subgraphs according to (M_i, S_i). A graphical representation of the XGDAG prioritization mechanism is shown in Figure 3.5.

Explainability methods for graph neural networks In our study, we made use of three XAI methods for GNNs. Each one of them relies on a different rationale to obtain explanation subgraphs. We will briefly remind their main characteristics, already described in Section 2.3. The first method is GNNExplainer [225]. It works by learning a mask on the adjacency matrix by maximizing mutual information. Its output is a subgraph of nodes that are relevant for the prediction (along with a subset of node features). Its predictions are edge-oriented. Another method we used

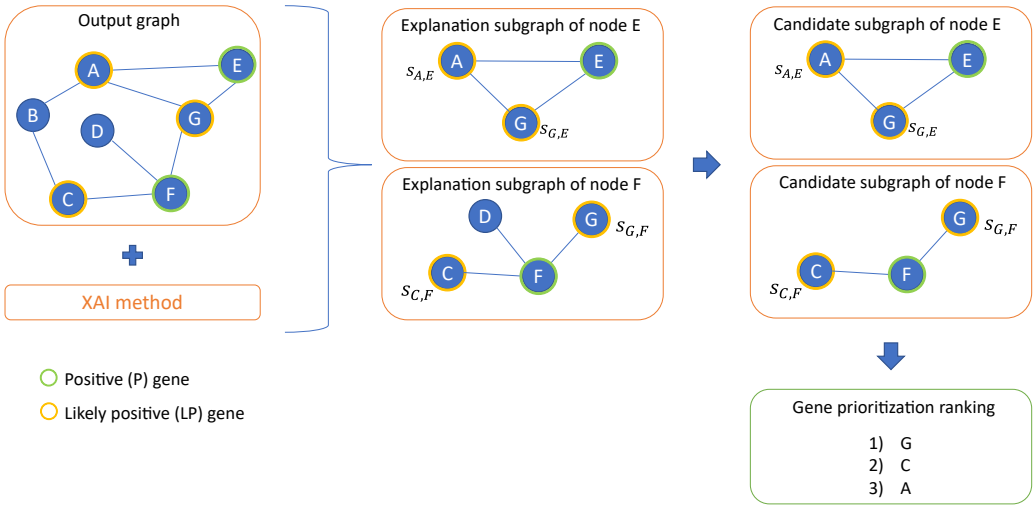


Figure 3.5. Graphical representations of the XGDAG prioritization mechanism. The output graph from the GNN is fed into an XAI method. For each P gene, we generate an explanation subgraph. This contains the nodes that were influential in the prediction of the node as P. We pool the subgraph by filtering out non-LP nodes, obtaining a final candidate subgraph. s_{ij} is the importance score assigned by a given explanation method to i for the prediction of node j. Assuming the cumulative importance score for node C to be greater than the one of node A ($S_C > S_A$), we obtain the gene ranking in the picture, with G as the top-ranked node because it appears in two candidate subgraphs.

is GraphSVX [240], which uses a surrogate model to approximate Shapley values [26] as indicators of node importance. This puts GraphSVX explanations on a solid and robust theoretical background. It delivers node-centric explanations. Finally, the third strategy we employed is SubgraphX [241], a subgraph-centric strategy that approximates Shapley values to determine the most relevant fully connected subgraph for predictions. The selected explanation subgraph is the one associated with the highest Shapley value. The three methods explain the predictions leveraging the three different key components of a graph: edges, nodes, and subgraphs, respectively. This allows us to have comprehensive explanations of the GNN predictions.

To use XAI methods as independent tools for prioritization for comparison purposes, we employ them in a PU learning setting. Indeed, we use them to explain models trained on binary PU data devoid of any prior label propagation. As a result, they lack the assistance provided by the classes generated during the label propagation phase, which can be considered as a preliminary prioritization. Without the aid of the LP class, the entire explanation subgraph is considered for prioritization without any node pooling. This introduces noise into the results and reduces the accuracy of the final ranking, as shown in Section 3.1.4 when comparing XGDAG-based variants

with standalone XAI tools. In more detail, for any node n, the G_n^{LP} set is absent in standalone XAI-based prioritization. Instead, we use the set G_n^U, which includes genes that are present in the explanation subgraph and that were predicted as unlabeled (U) by the GNN trained in the binary PU setting. Then, we proceed with the scoring and ranking criteria as proposed in Section 3.2.1.3. As mentioned earlier, using the entire set of genes predicted as unlabeled for prioritization introduces noise, as it may result in prioritizing genes that are highly unlikely to be associated with the disease, specifically the genes that would be predicted as RN by the GNN trained on the propagated labels. Conversely, the incorporation of label propagation in XGDAG brings additional value by facilitating the learning through pseudo-classes and assisting in the discovery of candidates through LP genes.

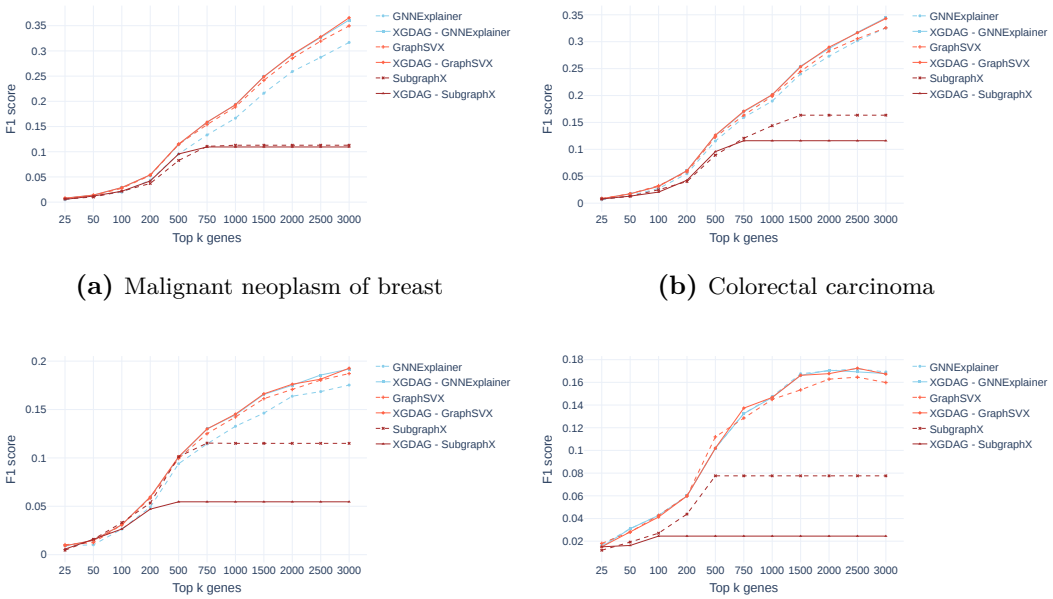
3.2.2 Results and Performance Analysis

To validate the obtained results, we performed both a numerical evaluation and an enrichment analysis. With the former, we compared, in terms of F1 score, the retrieval effectiveness of XGDAG with other methodologies for gene discovery; we compute the F1 score taking into consideration the number of associated genes in extended DisGeNET dataset that each method is able to detect. Seed genes present in the curated set are not considered for this purpose since they were used as positive genes for the training. This validation setting allows us to test whether our model was able to retrieve genes that had been discovered by previous research. In enrichment analysis, we inspected whether the set of genes prioritized by XGDAG was connected with the diseases under examination, namely whether the genes were enriched in pathways, gene ontologies, or other diseases associated with the considered ones.

3.2.2.1 Numerical Evaluation

First, in Figure 3.6, we compare the performance of XGDAG against the single XAI methods on which it is based, used as standalone tools. Notice that the PU learning-based XAI approach achieves higher performances with respect to its plain-explainability counterpart. Indeed, the use of the preliminary prioritization, obtained with the LP set from the label propagation phase, helps in the identification of the pool of possible new candidate genes.

We thus selected the best performing XGDAG variants in terms of overall F1 score. Given their at-par performance, we chose the GraphSVX- and the GNNExplainer-based approaches. The very similar behavior of the two strategies may be due to the fact that both GNNExplainer and GraphSVX identify important (and possibly over-



(a) Malignant neoplasm of breast

(b) Colorectal carcinoma

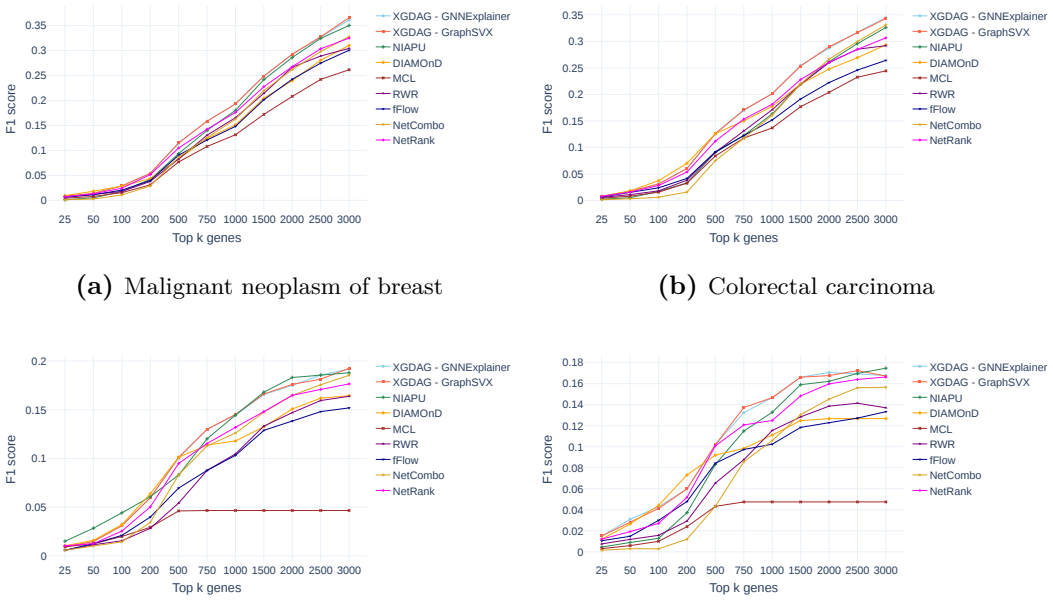
(c) Schizophrenia

(d) Depressive disorder

Figure 3.6. F1 score (y-axis) comparison for four representative diseases. The metrics are reported at increasing numbers of retrieved genes (x-axis). Dashed lines indicate the standalone XAI method and solid lines the XGDAG version. We notice that using explainability techniques on top of a PU learning prioritization strategy significantly improves the retrieval accuracy of the methods.

lapping) explanation subgraphs for the positive genes—noted also by inspecting the results of GNNExplainer and GraphSVX alone, yielding similar scores. Intersecting such explanation subgraphs with the LP set, effective prioritization is obtained. We thus compared them against state-of-the-art methodologies for gene prioritization, namely DIAMOnD, MCL, RWR, two variants of GUILD (fFlow and NetCombo), NetRank (based on the ToppNet algorithm of the ToppGene prioritization suite [61]), and also with NIAPU. The plots in Figure 3.7 show that XGDAG is more effective and robust than the other strategies. As we increase the number of retrieved genes, it is able to keep high the number of associated genes retrieved. On the contrary, methodologies such as DIAMOnD may be more effective in the retrieval when a small number of candidates are searched. However, they lose their reliability when higher numbers of candidate genes are considered, as also pointed out by DIAMOnD's designers [50] and previously in Section 3.1.4.3. In this, XGDAG proved to be the best solution even when looking for larger sets of candidate genes.

Results on a high-quality curated dataset By inspecting the results, we noticed the very high accuracy of DIAMOnD on small sets of candidate genes. The



(a) Malignant neoplasm of breast

(b) Colorectal carcinoma

(c) Schizophrenia

(d) Depressive disorder

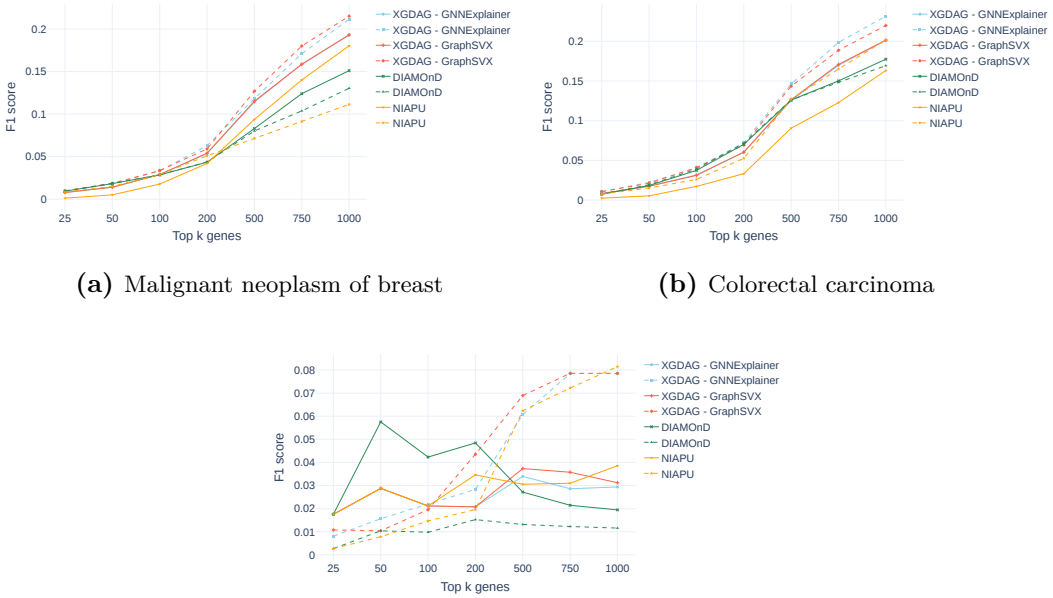
Figure 3.7. F1 score comparison for four representative diseases for the two best-performing XGDAG variants (GNNExplainer and GraphSVX) with known gene discovery methodologies. We notice that when the number of retrieved genes is small, the various approaches perform comparably. However, as the number of genes increases, XGDAG remains the most stable and robust method, whereas most of the compared strategies tend to become less accurate in the retrieval.

dataset we used, even in its curated version, contains a relatively high number of associated genes, some of them absent in other manually curated datasets. We were interested in exploring whether training on datasets with a higher level of curation and smaller numbers of associated genes would change these results.

We performed this additional experiment using the highly curated dataset by Ghiassian et al. [50]. This is the dataset on which DIAMOnD was trained and evaluated in the original publication. The PPI network used here was built considering physical interactions validated experimentally and gathered from different sources [308]. The GDAs were retrieved from OMIM (Online Mendelian Inheritance in Man) [309] and GWAS from PheGenI [310]. Because of the high-quality level of curation of these GDAs and PPI network, they were used in several gene prioritization experiments [260, 311, 312].

We used the PPI and the GDAs of the aforementioned dataset, which we call *OMIM+PheGenI* dataset, to train the algorithms. We then validated the mod-

els on the GDAs from the extended DisGeNET dataset. The goal was to first train the algorithms on high-quality and unbiased data and then test them on an external dataset. For this task, we considered the diseases in common between the two datasets: malignant neoplasm of breast (C0006142), colorectal carcinoma (C0009402), and liver chirrosis (C0023893). A comparative analysis of the F1 score is shown in Figure 3.8.



(a) Malignant neoplasm of breast

(b) Colorectal carcinoma

(c) Liver chirrosis

Figure 3.8. F1 score comparison for the OMIM+PheGenI dataset (dashed line) and the DisGeNET dataset (solid line). Even for a small number of genes, in this experiment XGDAG is competitive against DIAMOnD. The performances on the OMIM+PheGenI dataset are far superior to the DisGeNET ones.

The inspection of the results indicates that training on smaller but better-curated datasets is beneficial for XGDAG, whereas DIAMOnD suffers from training on smaller sets of seed genes. This further highlights the robustness of XGDAG whose results are accurate even when the number of seed genes is small. However, the different results obtained when using different datasets demonstrate that data quality plays a major role in gene discovery and prioritization tasks and that a particular focus should be put on the definition of high-quality GDAs and less biased interaction networks [301].

3.2.2.2 Enrichment Analysis

As a further analysis to enhance the validity of our methodology, we checked whether the candidate genes retrieved from XGDAG were enriched in biological pathways, gene ontologies, or other diseases related to the diseases of interest. We provide this analysis for the genes of the DisGeNET dataset prioritized by XGDAG-GNNExplainer. We considered the top 200 genes in our ranking as a reasonable cutoff. We performed the analysis using the Enrichr web tool [285, 286, 287] and selecting the most statistically significant results according to Fisher's exact test. For disease C0006142 (malignant neoplasm of breast) several significant gene ontologies and pathways were found. Figure 3.9 shows the ten most significant GOs for the biological process domain.

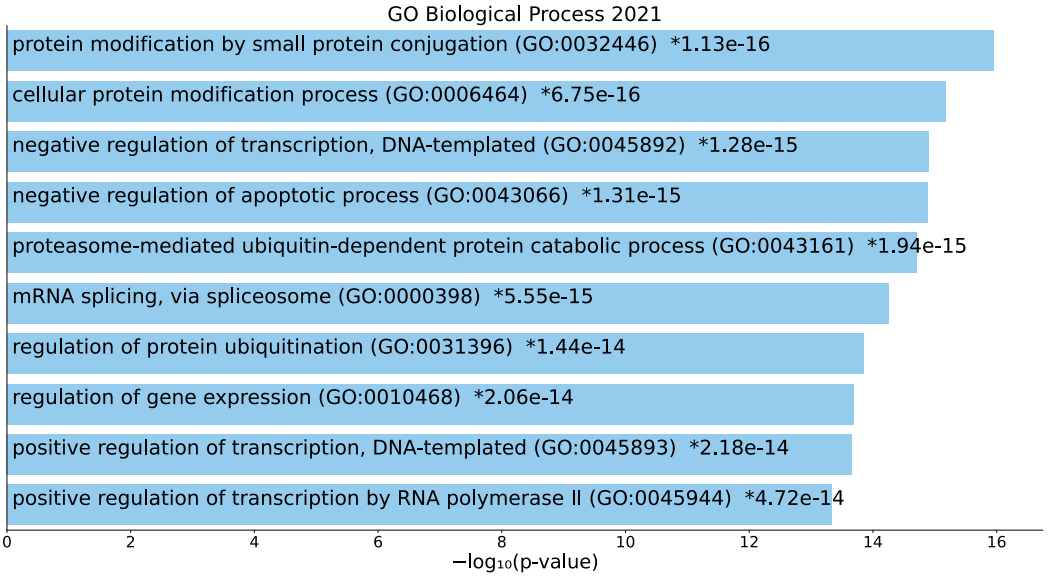


Figure 3.9. Top 10 significant gene ontologies for disease C0006142 (malignant neoplasm of breast) in the GO Biological Process 2021 database found with Enrichr. Breast cancer-related GOs are retrieved, further proving the effectiveness of XGDAG. Each item is statistically significant and reported with its p-value (*significant adjusted p-value).

Indeed, among the most significant GOs retrieved, protein modification was found to be a potential biomarker in breast cancer [313]. Moreover, dysregulated programs in DNA transcription are related to certain behaviors in cancer cells [314]. Furthermore, apoptotic process regulation plays an important role in cancer progression and therapies [315, 316, 317]. Enrichment analysis proved genes retrieved by XGDAG to have meaningful associations with the disease. Summarized results for the ten studied diseases providing the most enriched pathway, ontology, or associated disease and reference papers confirming the findings can be found in Table 3.5.

Table 3.5. Enrichment analysis for the considered diseases. We report the most enriched GO, pathway, or disease, with a description of the relationships and reference articles.

Disease	Enriched GO/pathway/disease	Relationships and references
C0006142 Malignant neoplasm of breast	GO:0032446 Protein modification by small protein conjugation	Protein modification was found to be a biomarker in breast cancer [313].
C0036341 Schizophrenia	Androgen receptor signaling pathway	Altered androgen receptor activity may impact stress in men with schizophrenia [318].
C0023893 Liver cirrhosis	GO:0042981 Regulation of apoptotic process	Apoptosis is a typical pathological feature of liver diseases and excessive apoptosis can generate acute liver injuries [319, 320].
C0009402 Colorectal carcinoma	GO:0006464 Cellular protein modification process	Protein synthesis deregulation is a frequent event in cancer, and many colorectal cancer mutations are responsible for the deregulation of translational processes [321].
C0376358 Malignant neoplasm of prostate	GO:0043066 Negative regulation of apoptotic process	The ability of cells to avoid apoptosis is crucial in cancer development and anti-apoptotic pathways play a major role in the development of effective treatments [322, 323].
C0005586 Bipolar disorder	Amyotrophic lateral sclerosis (ALS)	Hospitalized patients with bipolar disorder and psychiatric conditions were significantly associated with a first ALS diagnosis within a year [324].
C3714756 Intellectual disability	Neurodevelopmental disorder (Au-Kline Syndrome)	Au-Kline syndrome affects different body systems leading to intellectual disability, hypotonia, and delayed development [325].
C0860207 Drug-induced liver disease	Messenger RNA processing	A consistent number of circulating Messenger RNA and other microRNAs in plasma collected from drug-overdosed animals are found to be highly expressed in the liver [326].
C0011581 Depressive disorder	GO:0043066 Negative regulation of apoptotic process	Major depressive disorder shows evidence of local inflammatory, apoptotic, and oxidative stress [327].
C0001973 Chronic alcoholic intoxication	Dementia	Chronic abuse of alcohol can cause structural and functional brain damage, leading to alcohol-related dementia [328].

3.2.3 Observations

In this section, we presented a new methodology, XGDAG, which relies on PU learning, GNNs, and explainability to detect novel gene–disease associations by providing a prioritization of candidates. XGDAG uses the effective NeDBIT features defined in Section 3.1.3 to enable PU learning by assigning pseudo-classes to unlabeled instances via the NIAPU pipeline. This information is then leveraged by our GNN, which is able to generate network topology-aware embeddings that allow for high-accuracy predictions. In this context, accurate but black-box models do not provide any additional information than what we already know about gene associations. Thus, given that the reliability of the explanations will depend on the quality of the model itself, an accurate model is the base from which we start our explanation phase. The application of several XAI techniques (among which GNNExplainer and GraphSVX are the most effective) opens the black box on the GNN by determining the most influential nodes for the prediction. Some of these nodes are present in the set of genes predicted as LP: these nodes are selected as new candidate genes.

This is a novel use of XAI. Generally, the main goal of explainability is to gain insights into the decision process of a model. Diversely, in our approach, we exploit XAI methods to draw the final ranking of candidate genes, with the added value of having an explainable output. This is a novelty that presents XAI not only as a tool that opens the black box of deep neural networks but also as an analysis component directly incorporated into the GDA discovery pipeline tasked with producing the final output.

The method outperforms state-of-the-art methodologies for gene discovery, demonstrating the effective synergy of PU learning and explainability on GNN models. XGDAG's results are stable and robust, even considering large numbers of candidate genes.

It is interesting to point out that by using datasets with an in-depth level of manual curation, such as the one by Ghiassian et al. [50], the retrieval performance of XGDAG increases, demonstrating both the robustness of the approach and the importance of curated data.

Additionally, enrichment analysis uncovers associated pathways, ontologies, and traits linked to the selected diseases, backing up the accuracy of the gene ranking obtained with XGDAG and further proving its effectiveness as a gene discovery strategy.

Our approach is based on the analysis of general graph-structured data, so it can be used in various settings based on network modeling. It is thus possible to apply XGDAG on multiplex networks [41] and multi-omics data [40]. Notably, datasets such as the Omics Discovery Index [36, 37] and ConsensusPathDB [329, 330, 331] combine information from proteomics, metabolomics, genomics, and other interaction networks; expanding the study to encompass this type of data can further enhance the insights acquired through our methodology.

Finally, our study suggests that efforts can be put into the development of PU learning and XAI techniques devoted to GNNs for gene discovery purposes, giving the rewarding results that the joint use of such methods can obtain. The main limitation, as we observed in Section 3.2.2.1 is the requirement of high-quality data [301], also discussed in Section 3.1.5. This is shared by all data-based computational approaches; however, as more genes are discovered and validated, the results will be more trustworthy. The results of XGDAG were published in *Bioinformatics* by Oxford University Press [19].

Sections 3.1 and 3.2 were devoted to the development and analysis of techniques (NIAPU and XGDAG) to retrieve and prioritize candidate disease-associated genes. Moreover, we saw how NIAPU can ease learning for GNN models, enabling effective use of XGDAG via label propagation. In the next section, we will instead analyze possible interactions that can arise among those genes, the so-called epistatic interactions, and we will present EpiDetect, our explainable deep learning-based solution to detect them.

3.3 Explainable Deep Learning for Network Analysis of Epistatic Interactions

Differently from single gene–disease associations, traits or diseases in complex organisms may be regulated by the interaction of two or more genes. We are talking about epistatic interactions, introduced in Section 2.1.2. Two or more genes can interact, creating the phenomenon known as epistasis. The detection of epistatic genes is of extreme importance in genetics since it can help unveil mechanisms of diseases that are still unknown. Many computational methods have been developed for the purpose [71, 72, 74, 73, 75]. However, as introduced, those methods are negatively influenced by the presence of marginal (or main) effects. A main effect is the effect that a genetic variant alone has on the trait. Marginal effects can confuse a model that may detect as an epistatic interaction the presence of two independent,

strong contributions. In this thesis, we try to address this problem by developing a framework based on neural networks and explainability.

As remarked, despite their superior performance in various applications, their black-box character limits the use of deep learning models in genetics [4]: the complicated, highly nonlinear functions they learn make it difficult to understand how they operate, and thus render them unsuitable when it is needed to know why a particular output has been produced. Epistatic interaction detection is not immune to this.

To cope with this problem, we developed EpiDetect, a new framework for discovering potential epistatic interactions. The core element of EpiDetect is EpiCID (Epistatic Cosine Interaction Detection), a novel neural network-based algorithm that opens the black box and leverages the power of neural networks to discover complicated and hidden interactions between input single-nucleotide polymorphisms (SNPs), actually explaining the neural network predictions. EpiCID is a method specialized at directly detecting purely interacting input features without relying on the marginal effect that a single SNP may have on the trait under consideration. We remind that SNPs are genetic variants occurring when a nucleotide (A, C, G, T) is substituted with another nucleotide in a given gene or DNA region (locus). Our method is designed to deliver global explanations. In contrast with most methodologies, including the other XAI strategies presented in this thesis, which are local instance-based explainers that discover critical features for specific predictions, EpiCID determines the impact (in terms of interaction strength) that pairs of features have globally on the behavior of the model.

Given the high relevance and impact of the research in the field, we selected blood pressure regulation as a case study [332]. We designed three regression models based on an MLP for detecting high systolic (SBP), diastolic (DBP), and pulse pressure (PP), and we evaluated the weights of the layers inside the neural networks and the effect that these weights have on the final output (the blood pressure value). For each pair of SNPs (our method can also be extended to subsets of more than two SNPs), we obtained an interaction score, which captures the strength of the interaction among the two SNPs of the pair, as well as the effect that this interaction has on the final regression function. Then, each SNP is mapped to the corresponding gene. Subsequently, we identified the genes/SNPs with the most interactions by creating a network and connecting those with a high interaction score. From this network, we identified the most central genes that significantly affect SBP, DBP, or PP. This allowed us to discover potentially novel pathways that affect blood pressure and cardiovascular risk. To compare our approach, we evaluated it against

other widely used epistasis detection frameworks such as the mentioned Multifactor Dimensionality Reduction (MDR) [71] and Boolean Operation-based Screening and Testing (BOOST) [72]. We also compared it with Neural Interaction Detection (NID) [223], a methodology specific to neural networks that we adapted to work with genetic inputs. Our results showed that our strategy outperformed these methods and minimized the influence of marginal effects.

We hereby present our proposed framework. Given subject patients' genotype (SNPs) and phenotype (blood pressure value), our system identifies genes correlated with the phenotype expression based on their interaction. EpiDetect consists of three main components: i) the EpiCID neural network explainability module that detects candidate epistatic pairs, ii) the design of a gene–gene network and centrality analysis, and iii) enrichment analysis. We present a graphical representation of the workflow of our approach in Figure 3.10.

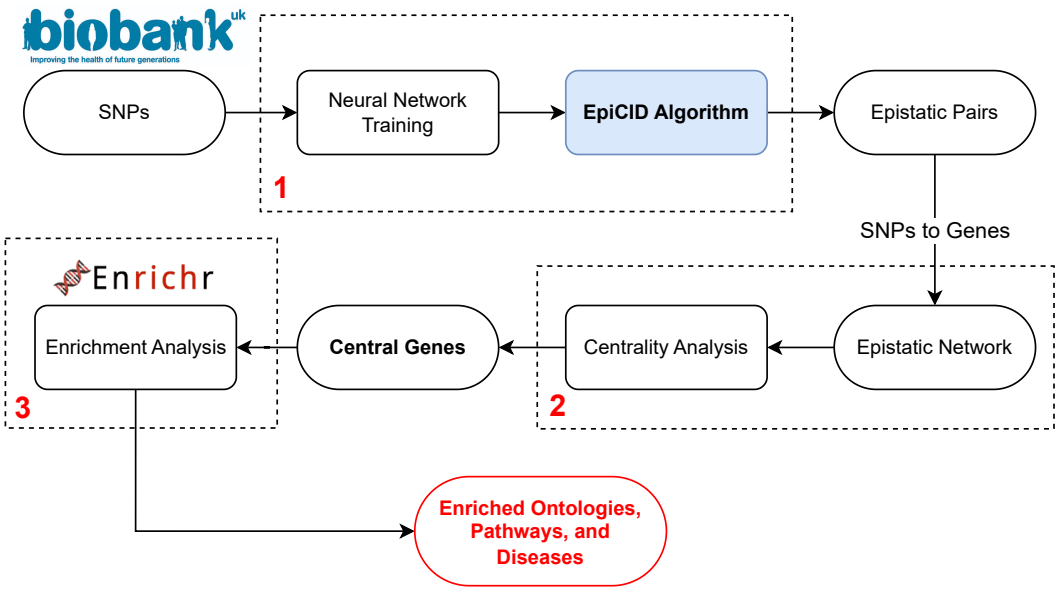


Figure 3.10. The workflow of the EpiDetect framework. The numbers in the blocks indicate the three components of the pipeline. A neural network is trained and explained with EpiCID (block 1) to find interacting pairs that are mapped to genes and used to build an epistatic network and perform centrality analysis (block 2). The central genes are used for enrichment analysis (block 3), which determines associated ontologies, pathways, and diseases.

We applied our EpiDetect framework to the three blood pressure traits. Herein, we report the results obtained and compare them to those of the other approaches (MDR, BOOST, and NID). In Section 3.3.1, we describe how we gathered and

processed genetic information. Then, in Section 3.3.2, we delve deeper into the methodology, describing the neural network models and the EpiCID algorithm (block 1 of Figure 3.10), the network and centrality analysis (block 2), and the final enrichment analysis (block 3). The complete EpiDetect pipeline provides an output at different levels, from lists of interacting SNP pairs to central genes and biological pathways. Therefore, in Section 3.3.3, we perform a three-level analysis: first, we describe the results of the centrality analysis, which recovered the most promising genes from the EpiCID ranking, and then we delve into the results of the enrichment analysis phase to understand the biological relevance of our findings. Finally, we go back to EpiCID to analyze its output ranking and evaluate its robustness.

3.3.1 Data Curation

For our study, we retrieved individuals' genotype and phenotype information from UK Biobank [333, 18], a large population-based cohort in the UK that includes around half a million volunteers aged 40 to 69 years. For our analysis, we selected SNPs that were found to be robustly associated with SBP, DBP, and PP, taking into account genome-wide significant signals for those traits derived from the largest GWAS on blood pressure [332]. This allows us to narrow down the pool of possible epistatic SNPs, reducing the complexity associated with those studies. Thus, we ended up having 264 SNPs for SBP, 342 for DBP, and 283 for PP. We created three datasets, one for each trait, in which an array of SNPs describes each individual, and the target variable is the blood pressure value of interest. We calculated the mean SBP, DBP, and PP values from two automated or, wherever needed, manual measurements. We further performed quality control, excluding individuals with more than 10% of missing genotype. After the quality control, we ended up with 456,057 individuals. Every patient in the three datasets is described by an array of SNPs. The SNPs were encoded in a one-hot encoding fashion to be suitable to be fed to a neural network model. Each SNP is a three-element vector in which the one-valued element indicates the genotype:

- $[1, 0, 0]$: homozygous for the major allele.
- $[0, 1, 0]$: heterozygous.
- $[0, 0, 1]$: homozygous for the minor allele.

The major allele is the most frequent in the population, while the minor is the least occurring.

3.3.2 Methodology

As introduced, EpiDetect is composed of three main elements. The first one, EpiCID, takes in input a trained neural network and outputs interacting pairs of SNPs/genes. The second component, the network analysis, retrieves the most important genes in the epistatic network and finally, the enrichment analysis, looks for pathways or ontologies associated with the disease or trait under scrutiny. We will analyze each component of the framework in detail.

3.3.2.1 Neural Network Model and Training

Our methodology is based on the usage of an MLP. The use of this simple type of neural network is aligned with the necessity in genetics to develop simple and explainable models rather than complex and obscure ones [197]. Our network is composed of two fully connected layers with 200 and 50 neurons each, followed by a dropout layer (probability of dropout set to 0.3). For each blood pressure trait (SBP, DBP, and PP), the network was trained for 40 epochs using Adam optimizer [307] with a learning rate of $1e-4$ and a batch size of 16, generating three different models. The three blood pressure datasets were split into training (70%) and test sets (30%). The average mean absolute error on the test set of the networks was around 10 mmHg (the unit of measurement for blood pressure). The trained neural network is the base for the explanation phase. As a note, even though the regression error may seem high, it is important to point out that, with this study, we are capturing only the genetic components of blood pressure regulation without taking into account other important factors like patients' habits and lifestyles that have an impact on the traits.

3.3.2.2 Epistatic Cosine Interaction Detection

The core of EpiDetect is the Epistatic Cosine Interaction Detection (EpiCID) algorithm. Our idea is that to find interacting pairs of input features, it is possible to consider their correlation in the vector space. Thus, one possible metric to employ in order to measure feature correlation is cosine similarity. From this follows the need to represent a feature as a vector, using a sort of "feature vector of a feature". This vector should describe how the feature is represented and internally seen by the neural network. It can be thought of as a fingerprint or embedding of the feature itself. We can build this vector by using the weights the neural network learned during its training. To construct this vector, we consider the first-layer weights (the ones connecting the input with the first-layer neurons) and the influence that hidden units exert on the output. The latter can be taken into account using the concept of *aggregated weight* introduced by NID authors [223]. This is computed by cumulative

matrix multiplications of the absolute values of weight matrices and indicates the influence hidden units have on the output. The aggregated weights of the units at layer l are defined as

$$\mathbf{z}(l) = |\mathbf{w}^y|^T \cdot |\mathbf{W}^{(L)}| \cdot |\mathbf{W}^{(L-1)}| \dots |\mathbf{W}^{(l+1)}|, \tag{3.5}$$

where $\mathbf{w}^y$ are the weights at the output layer, $\mathbf{W}^{(j)}$ is the weight matrix at hidden layer j and L is the network depth (i.e., the number of hidden layers). Thus, each element $z_k(l)$ indicates the aggregated weight at unit k of layer l. As demonstrated [223], this definition is an upper bound of the gradient magnitude. It can be used as an approximation of the importance of the hidden units, as gradients have been commonly employed as importance measures in neural networks [220, 334, 217], as also indicated in Section 2.3. Now that we have all the elements needed, we can define the *neural feature vector* $\xi^{(i)}$ for input feature i as follows:

$$\xi^{(i)} = |\mathbf{W}_i^{(1)}| \odot \mathbf{z}^{(1)}, \tag{3.6}$$

where $\odot$ indicates the Hadamard product of two vectors, $\mathbf{W}_i^{(1)}$ is the vector of the weights between feature i in the input and the first hidden layer units, and $\mathbf{z}^{(1)}$ is the aggregated weight vector at the first hidden layer, as defined in Equation (3.5). Every element $\xi_k^{(i)}$ is given by the product between the connection weight from feature i to unit k of the first hidden layer and the aggregated weight at such unit. The vector defined above is an embedding of the feature, a hidden representation created by the network during the learning process, implicitly considering its contribution to the output throughout the network.

Cosine interaction strength Now, we know how to represent a feature in the vector space as embedded by the neural network. We can exploit this to compute the interaction strength between two or more features. We decided to consider the correlation of the feature in the vector space as a measure of interaction importance using the cosine similarity measure, defining the *cosine interaction strength* between features i and j as

$$\phi(i,j) = \cos(\xi^{(i)}, \xi^{(j)}) = \frac{\xi^{(i)} \cdot \xi^{(j)}}{\|\xi^{(i)}\| \, \|\xi^{(j)}\|}, \tag{3.7}$$

where $\xi^{(i)}$ and $\xi^{(j)}$ are the neural feature vectors describing features i and j. Notice that $\phi(i,j) \in [0,1]$ since $\xi_k^{(i)} \geq 0$ and $\xi_k^{(j)} \geq 0$ for every k, as per Equations (3.5) and (3.6). The rationale behind this measure is that a high cosine similarity between two neural feature vectors indicates that two features interact in the network. However, this measure alone does not capture the real magnitude of the interaction, but it is only able to detect if an interaction is present. Early experiments showed this measure could find interacting pairs of features but failed to determine the interaction strength properly. For that reason, we improved the definition of cosine interaction strength, defining a more appropriate one that was able to differentiate between weak and strong interactions.

In order to do that, we took into consideration where the interaction is created, i.e., between the input and the first hidden layer. Inspired by NID, given features i and j, we consider their connection weights to the first hidden layer units. Differently from Equation (3.6), in which the effect of a single feature on the network is considered, we now analyze the joint effect of the two features on the first hidden layer. For each first-layer unit k, we apply the min averaging function to the absolute values of the weights connecting k to i and j. The choice of this function is justified by previous work [223], which demonstrated its effectiveness in detecting feature influence in neural networks. Then, we sum all over the units to obtain what we defined *first-layer interaction influence*:

$$\eta(i,j) = \sum\nolimits_k \min(|W_{i,k}^{(1)}|, |W_{j,k}^{(1)}|). \tag{3.8}$$

The rationale behind the choice of the minimum is that an interaction is strong (at the first hidden layer) when its η is large. When the minimum of the two weights is large (i.e., when both weights have a high value) for a consistent number of hidden units, η will be high, indicating the rise of strong interaction. Conversely, the previously defined ϕ in Equation (3.7) helps understand how the interaction evolves when passing throughout the network up to the output; an interaction may start strong at the first hidden layer but lose its importance toward the output layer, or a mild interaction may preserve its strength, having a relevant impact on the

prediction. Given those observations and merging Equations (3.7) and (3.8), we obtain the new definition of cosine interaction strength:

$$\Phi(i,j) = \eta(i,j)\phi(i,j). \tag{3.9}$$

The measure defined in Equation (3.9) can describe the interaction completely and coherently, from its rise from the input to the first hidden layer and considering its evolution toward the output through the network.

Finally, in our application, given that a SNP is represented by a three-element one-hot-encoded vector (see Section 3.3.1), it follows that it is described by three different neural feature vectors (one for each feature), which need to be aggregated before computing the interaction strength. We merged them into a single one by performing an element-wise sum of the vectors. The same aggregation was used for the first-layer weights to compute the first-layer interaction influence of a SNP. Once we have the vectors, it is possible to obtain the cosine interaction strength for a pair of SNPs as in Equation (3.9). Notably, EpiCID can be extended to sets of more than two SNPs.

3.3.2.3 Network and Centrality Analysis

After having trained the MLP three times (once for each trait; SBP, DBP, and PP) and applied EpiCID, we obtained a ranked list of highly interacting (as measured by EpiCID) pairs of SNPs. We next mapped these SNPs to genes, obtaining a ranking of interacting genes. The mapping is performed via the following steps:

1. We map a SNP to a protein-coding gene according to the Consensus Coding Sequence Project (CCDS) [335], which provides very high-quality annotations for protein-coding genes.
2. If no reference in CCDS is found, we map using Genecode [336].

In cases where a SNP is annotated in a genome region with more than one gene present, we followed the process:

1. We map to a coding gene if present.
2. If more than one coding gene is present, we choose a mapping already appearing in the literature.

3. If no coding genes are present, we look up different sources, such as dbSNP [337] and Ensembl [338], to choose the most suitable mapping.

Whenever we do not find a mapping, we assign the SNP to the closest gene according to Genecode, giving a preference for coding genes.

We used the interactions identified by EpiCID to create an epistatic interaction gene–gene network for each of the three blood pressure traits and discover the most central genes by studying the top 1000 interactions [339, 340]. For comparison purposes, to investigate the effectiveness of the proposed methodology, we also analyzed the central networks obtained by using the other algorithms for epistatic interaction detection. The central genes, i.e., those with a degree higher than the average degree in the top-1000 network, are chosen for enrichment analysis. The degree, corresponding to the number of direct neighbors to a given node, allows an immediate evaluation of the regulatory relevance of a node and can be used to validate centrality in different kinds of networks, such as signaling and metabolic networks [341].

3.3.2.4 Enrichment Analysis

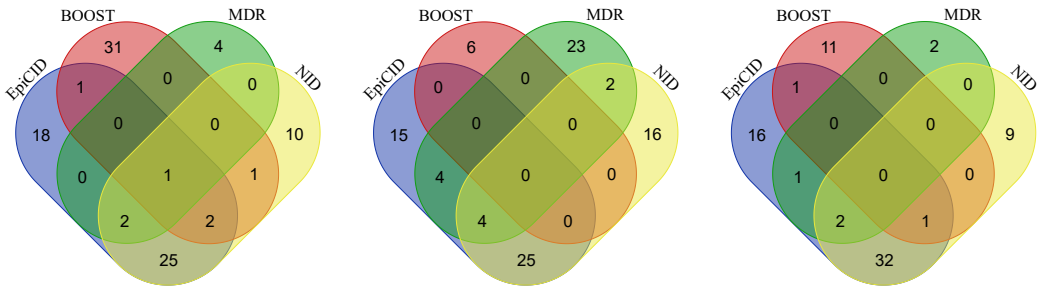
We performed enrichment analysis for central genes of the epistatic networks using again the Enrichr online tool [285, 286, 287]. We compared gene sets from each algorithmic approach (EpiCID, MDR, BOOST, and NID) to the datasets available in the databases of the Gene Ontology Consortium [342], DisGeNET discovery platform [262, 48, 14], GWAS Catalog [343], and UK Biobank [333, 18], to understand the biological relevance of the epistatic networks obtained for the three blood pressure traits.

3.3.3 Analysis of the Results

We hereby present the results of the application of the EpiDetect pipeline. Given that validating epistatic interactions is rather challenging due to the complex nature of the phenomenon and the absence of consistent ground truth, we proceed with a three-fold analysis to have a thorough evaluation of our methodology. We will first describe the network and centrality analysis of the interacting pairs found by EpiCID and compare them with results obtained with other algorithms (Section 3.3.3.1). Then, we delve into the enrichment analysis, which provides important biological insights (Section 3.3.3.2). Finally, Section 3.3.3.3 is dedicated to the analysis of the EpiCID output to investigate its reliability and robustness, mainly focusing on marginal effects and interaction distributions.

3.3.3.1 Centrality Analysis Results

The top 1000 interactions provided by EpiCID involved 195, 168, and 185 interacting genes, while centrality analysis resulted in 49, 48, and 53 central genes for the SBP, DBP, and PP traits, respectively. Next, we looked for common genes among the EpiCID-derived network and the networks obtained from BOOST, MDR, and NID (Figure 3.11).



(a) SBP **(b)** DBP **(c)** PP

Figure 3.11. Common central genes from SNPs associated with SBP **(a)**, DBP **(b)**, and PP **(c)**, following analysis with different algorithms.

Thirty-one out of 49, 33 out of 48, and 37 out of 53 central genes obtained by EpiCID for SBP, DBP, and PP, respectively, were common to the central genes obtained by at least one or more other approaches. EpiCID had the most common genes with NID, compared to other methods, namely 30, 29, and 35 for SBP, DBP, and PP, respectively. MDR contained an intermediate but low number of common central genes with EpiCID and the other strategies, 3 out of 7, 10 out of 33, and 3 out of 5. BOOST had the least common genes to the other three approaches; we found 5, none, and 2 common genes for SBP, DBP, and PP, respectively. EpiCID led to a high percentage of common central genes with methodologies adopting similar architecture (i.e., NID), whereas BOOST and MDR led to more divergent sets of central genes.

3.3.3.2 Enrichment Analysis Results

Centrality analysis led to a gene set that contained only a fraction of the total genes of the epistatic network, which are assumed to play a central role in the trait/disease network under consideration. A critical question is which epistatic network may more accurately represent the affected intracellular pathways that alter organismal physiology and lead to a trait/disease. In the enrichment analysis, we queried for the traits that are most significantly associated with each of the sets of central

genes returned by the four approaches (EpiDetect, NID, MDR, and BOOST) as a means to discover which method returns a core network for each trait. So, we examined whether SBP, DBP, and PP traits were found as associated traits and, if yes, in which position in the enrichment ranking; we report the position for the different approaches in Table 3.6. From the ranking of the Enrichr results, we can observe that the networks obtained by EpiDetect can represent core networks on each associated trait more accurately compared to the other algorithms. With the exception of the PP trait in DisGeNET and UK Biobank, where none of the algorithms predicted the trait, EpiDetect-associated traits were ranked first in all cases. NID had slightly lower accuracy, with DBP in UK Biobank ranking third. Finally, MDR and BOOST had the lowest accuracy among the methods tested, as the relevant traits were ranked in lower positions. This difference can be quantified by a Point Penalty Score that we defined, which penalizes the approaches that perform worst (see Table 3.6) considering the ranking among the proposed strategies. The method ranking the trait higher than the others gets 0 penalty points, the second method gets 1 point, and so on (ties get the same penalty score). A lower score is better. No penalty is assigned if no method ranks the trait among the top 10 enriched terms. EpiDetect scores 0, being always the top-performing approach able to retrieve gene networks representative of the analyzed disease.

Table 3.6. Association of central genes to the respective trait (N/A: not associated among the top 10 terms). We also present the results of our framework when we substitute the first component (EpiCID) with BOOST, MDR, and NID for comparison. The rank indicates the position of the trait in the retrieved ranking of traits/diseases associated with the central genes of each algorithmic approach in each of the three databases. Each method is presented with its Point Penalty Score.

Algorithm	**Rank**									**Point Penalty Score**
	DisGeNET			**GWAS Catalog**			**UK Biobank**			
	SBP	**DBP**	**PP**	**SBP**	**DBP**	**PP**	**SBP**	**DBP**	**PP**	
EpiDetect	1	1	N/A	1	1	1	1	1	N/A	0
BOOST	1	1	N/A	1	1	1	1	N/A	N/A	2
MDR	5	1	N/A	3	1	N/A	3	1	N/A	4
NID	1	1	N/A	1	1	1	1	3	N/A	1

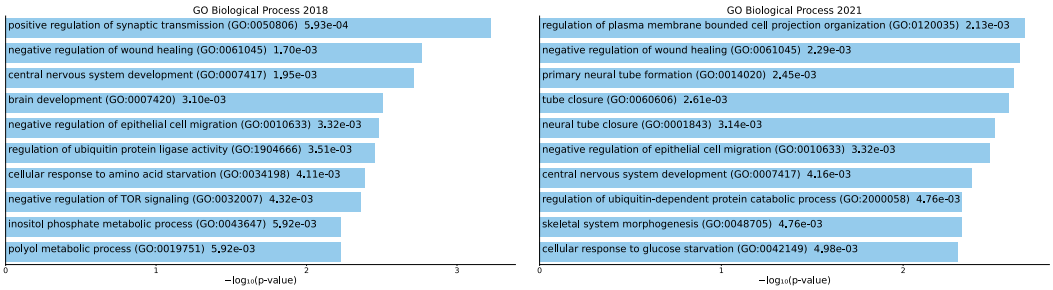
Having verified the specificity of EpiDetect, we next analyzed for enriched gene ontologies (GO Biological Process databases) found using the central genes of the epistatic networks. We performed a ranking of the pathways found, based on the obtained p-value, following Fisher's exact test; the top results are shown in Figure 3.12.

The enrichment of the SBP network revealed associations with GOs related to the regulation of synaptic transmission (GO:0050806) and other nervous system-related ontologies (GO:0014020 and GO:0001843, among others), in line with the evidence of its connection with blood pressure regulation [344, 345], as well as established associations between hypertension and wound healing processes (GO:0061045) [346] and plasma membrane abnormalities (GO:0120035) [347]. DBP-central genes were enriched in MAPKs and MAPK signaling (GO:0051403 and GO:0031098) pathways known to be associated with blood-pressure traits [332]. A novel finding is that cellular sodium ion homeostasis (GO:0006883) and similar GOs (GO:0055078 and GO:0030004) were found enriched based on EpiDetect epistatic network genes. In the PP network of central genes, among the known associations were that of regulation from RNA-polymerase II (GO:0045944) [348] and cardiac muscle fiber development (GO:0048739) [332]. Also, sarcomere organization (GO:0045214) was found to be enriched and in line with previous studies [349]. Among others, an interesting association is that of the Notch signaling pathway (GO:0008593). In summary, EpiDetect-derived epistatic networks provide supporting evidence for known pathways associated with blood pressure traits, as well as the potential for discovering novel pathways.

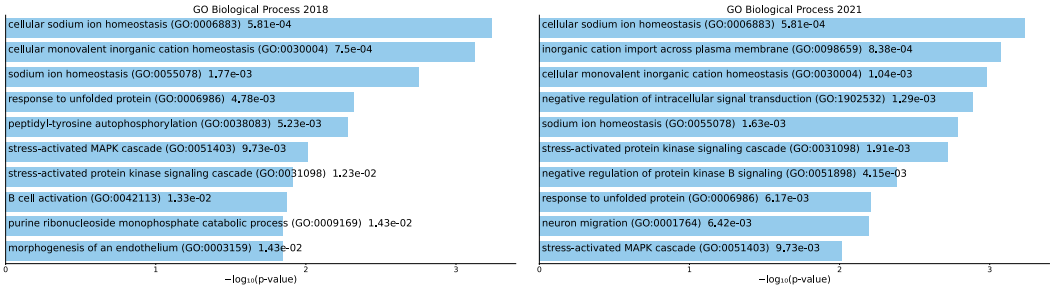
3.3.3.3 Marginal Effect Analysis

A complete evaluation of our results involves the entire pipeline of the EpiDetect framework. Yet, here, we want to investigate possible bias in the output rankings of the various algorithms for epistasis detection caused by the presence of marginal effects. We can notice this bias by looking at the rankings and examining whether they are dominated by the influence of a small number of SNPs that appear to interact with an exceedingly high number of other SNPs; such a result would be quite unusual in a ranking with nonlinearly interacting pairs [339]. By inspecting the degree of the 5 most interacting SNPs in the top-1000 interaction network for SBP (Table 3.7), we notice how in the BOOST and MDR output only a small number of SNPs is involved in almost all the top-1000 interactions.

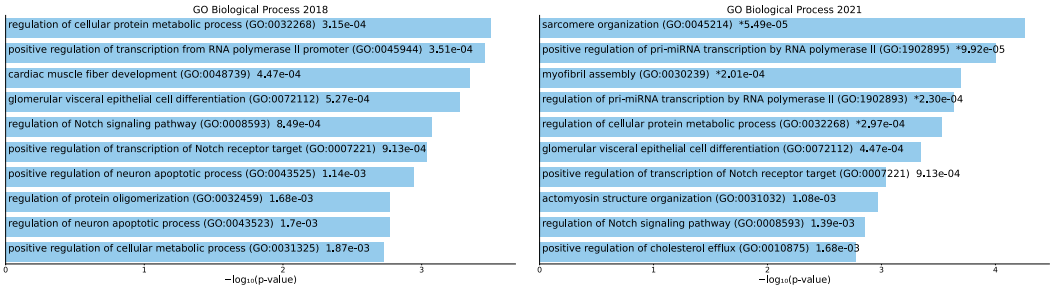
For BOOST, 5 SNPs are involved in 782 of the top 1000 interactions. This concentration is much more evident with MDR, with 972 interactions. In contrast, the top 5 SNPs are involved in 575 interactions for NID and even less for EpiCID (466). Thus, neural network-based methods, especially EpiCID, exhibit more variability in the distribution of the interacting SNPs. This suggests that these methodologies are less affected by the marginal effects of single SNPs. We can observe this behavior also in Figure 3.13.



(a) SBP



(b) DBP



(c) PP

Figure 3.12. Gene Ontology Biological Process enriched terms (databases 2018 and 2021) in EpiDetect central genes from SBP **(a)**, DBP **(b)**, and PP **(c)** epistatic networks. Every term is statistically significant and reported with its p-value (*significant adjusted p-value).

Next, we compared the different algorithms with respect to the distribution of the SNPs with the highest number of interactions at different levels. Figure 3.14 shows stacked bar charts visualizing the most interacting SNPs in the top-100, top-500, and top-1000 interaction networks. In the first 100 interactions, BOOST is characterized by only one SNP interacting with all the rest of the SNPs, suggesting a presence of bias toward this particular SNP, presumably caused by a strong marginal effect (Figure 3.14a). MDR has just 3 SNPs dominating the top 100 interactions, and almost 80 interactions show the presence of the same SNP. On the contrary, for NID and EpiCID, 5 and 6 different SNPs are involved in the top 100 interactions, respectively.

Table 3.7. Distributions of the 5 most interacting SNPs in first 1000 interactions for SBP (the number of interactions corresponds to the degree in the top-1000 interaction network) and the total number of interactions in which they are involved.

BOOST			
Rank	**SNP**	**Gene**	**Interactions**
1	rs1012089	AC138627.1	226
2	rs9667596	OR4A44P	180
3	rs7023828	AL358074.1	157
4	rs10224002	PRKAG2	130
5	rs1694068	ARL15	89
Total number of interactions			782
MDR			
Rank	**SNP**	**Gene**	**Interactions**
1	rs17477177	AC004917.1	263
2	rs17249754	ATP2B1	263
3	rs11191548	CNNM2	237
4	rs1173771	NPR3	163
5	rs17367504	MTHFR	46
Total number of interactions			972
NID			
Rank	**SNP**	**Gene**	**Interactions**
1	rs77413490	PTEN	163
2	rs1126930	PRKAG1	115
3	rs7331680	CDC16	114
4	rs28621435	GRIN2B	92
5	rs139354822	FARP2	91
Total number of interactions			575
EpiCID			
Rank	**SNP**	**Gene**	**Interactions**
1	rs77413490	PTEN	150
2	rs1126930	PRKAG1	118
3	rs75961402	HNF4GP1	68
4	rs7331680	CDC16	66
5	rs10437954	ARHGEF25	64
Total number of interactions			466

We obtained similar insights with the top 500 interactions (Figure 3.14b). For BOOST and MDR, we found 4 top-interacting SNPs, 10 for NID and 14 for EpiCID. This is further confirmed by the top-1000 chart (Figure 3.14c), which shows 9 top SNPs for BOOST, only 6 for MDR, 14 for NID, and 20 for EpiCID. Given the tendency of standard algorithms to be more strongly affected by marginal effects, they may wrongly detect spurious interactions as pure ones, impinging on the output

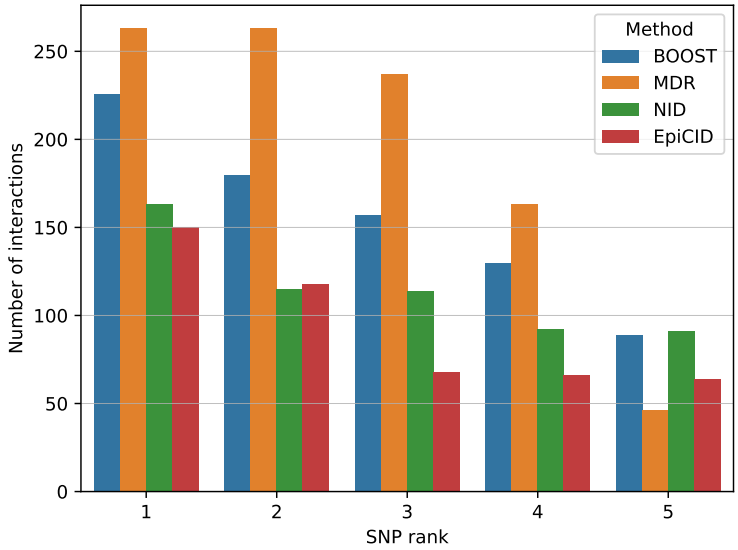
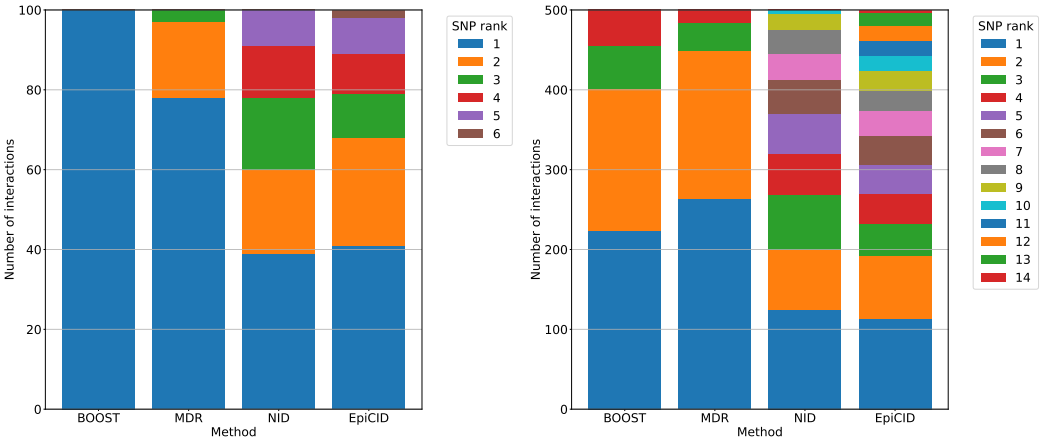


Figure 3.13. Graphical representation of the distribution of the 5 highest-degree SNPs in the first 1000 interactions for SBP.

and resulting in unusual rankings with small numbers of SNPs, also reflected in the centrality analysis. On the contrary, neural networks-based models, especially EpiCID, returned a more variable output in the number of interacting SNPs, which is what we would expect in nature, probably providing a more accurate depiction of several interactions that underlie complex traits [350]. Analogous results were obtained for DBP and PP, available in Appendix A.

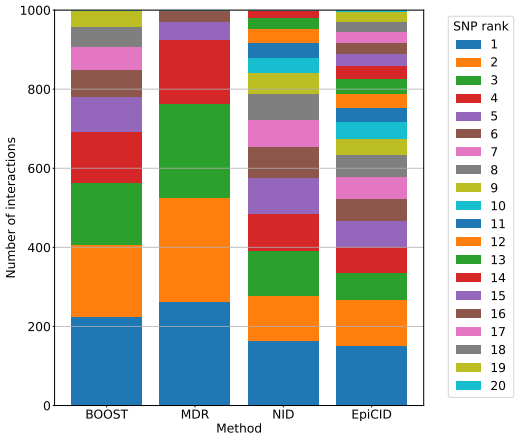
3.3.4 Observations

In this section, we have seen that complex traits are typically influenced by multiple genetic factors, making their analysis challenging. We thereby developed EpiDetect, a novel explainable deep learning-based method, and we provided a step-by-step protocol to detect epistatic effects and create epistatic gene networks. Our approach detected meaningful epistatic interactions, delivering results that are more accurate than the ones of the other approaches that we compared with (MDR, BOOST, and NID). EpiDetect consists of three major components. First, EpiCID, a novel algorithm that calculates epistatic interactions relying on the inner mechanics of neural networks, properly unveiled by our explainability scheme. Next, a network analysis component is used to extract central genes. Finally, we apply an enrichment analysis process to those central genes. The framework provides a three-level output: ranked interactions of SNPs, central genes, and pathways/ontologies. This



(a) Top 100 interactions.

(b) Top 500 interactions.



(c) Top 1000 interactions.

Figure 3.14. Highest-degree SNPs in the top 100 **(a)**, top 500 **(b)**, and top 1000 **(c)** interactions for SBP. On the x-axis, we have the method for detecting interactions, and on the y-axis, the number of interactions in which the most popular SNPs are involved (corresponding to the degree in each top-n interaction network).

multilevel output helps provide a more differentiated view of the results, that would be otherwise hard to evaluate given the complexity of the epistasis phenomenon and the lack of a consistent ground truth.

The first component of the framework identifies purely interacting features, minimizing marginal effects of single SNPs, which can lead algorithms to detect spurious interactions; this represents a novelty even in the field of neural network explainability. Interaction detection approaches usually rely on importance metrics of single features to compute interaction scores [351]; this may leave out interactions for which the main effects are negligible. EpiCID exploits the neural network weights

learned and optimized during the training process and, by extracting feature embeddings in terms of the newly defined neural feature vector, is able to determine how an interaction is created and evolves in the neural network, and its effect on the output. The result of the EpiCID component is a list of ranked epistatic interactions.

The second component is a network analysis workflow, which calculates a gene–gene network based on the top-ranked SNP–SNP interactions previously obtained. This process is in agreement with the concept that a holistic, systems biology approach is needed to explain gene–gene interactions, which explain epistasis and complex phenotypes [352, 353, 354, 355]. It uses centrality analysis to calculate the most significant/central genes in the EpiCID-derived networks [356]. Central genes were identified using the degree measure, which was proven to be effective in the analysis of biological networks. A protein with a high degree interacts with several other proteins, suggesting a possible central regulatory role. There is convincing evidence that proteins with a central regulatory role, or hubs, have been elucidated to be central oncogenes or tumor suppressor genes with clinical significance in cancer-related networks [357, 358, 359, 360]. However, our workflow can be perfectly adapted to use other notions of centrality measures [361]. We validated the proposed approach by repeating the workflow using a number of epistatic analysis algorithms, namely MDR, BOOST, and NID. The accuracy of our strategy was assessed by comparing it with the final central gene networks from the other methods. In all three traits (SBP, DBP, and PP), EpiCID had more common genes with NID, whereas MDR and BOOST networks featured more unique genes. EpiCID resulted in a large set of shared central genes with the compared strategies. The fact that EpiCID identified genes that were also found by other methods highlights its robustness. At the same time, the discovery of unique genes found using EpiCID may be more reliable and likely to represent the actual gene networks underlying these traits (confirmed by the subsequent enrichment analysis). This may indicate that it might depict the underlying regulatory network more accurately, rendering the unique genes found more reliable. The common, as well as the unique genes found by EpiCID, might have a significant role in the actual mechanism of pathogenesis. In that sense, the results we observed with EpiCID suggest it may represent a more accurate depiction of the "biological truth."

The third component of EpiDetect is the enrichment analysis using Enrichr to explore how accurately our approach depicted the core gene regulatory network underlying each trait. Central genes obtained with EpiCID performed similarly to NID, whereas MDR and BOOST lacked accuracy in the association with the respective traits. Interestingly, the degree of common genes between EpiDetect

and NID was also informative of their performance after enrichment analysis. Both approaches had the best performances. However, EpiDetect was slightly more accurate in predicting the associated traits. These results also indicate the advantage of neural network approaches compared to classical statistical methods in predicting epistatic interactions that lead to complex traits. Further pathway analysis of EpiDetect-derived central genes offered insight into the possible underlying mechanisms of complex traits. GO analysis revealed both known and novel pathways associated with SBP, DBP, and PP. Regarding SBP, associations with ontologies related to the nervous system, such as synaptic transmission, brain development, and neural tube formation and closure, were found. This is in line with previous studies with evidence of increased activation of the central nervous system as a contributor to hypertension [345] and impact of brain and nervous system development on blood pressure regulation [344]. Further enriched terms are confirmed by research linking hypertension with delayed wound healing [346] and plasma membrane defects [347]. MAPK signaling was found to be associated with DBP central genes. This result replicates previous studies [332] and provides further evidence of the significance of stress-associated p38-MAPK signaling in hypertension in experimental models of hypertensive mice and rats [362, 363]. A novel finding was that cellular sodium ion homeostasis was found to be associated with DBP. The latter may explain secondary hypertension in individuals suffering from other pathologies, such as aldosterone-producing adenomas [364] and kidney disease [365]. Regarding PP, regulation from RNA-polymerase II [348] and cardiac muscle fiber development [332] are GOs with an established role in PP that were found to be associated with PP EpiDetect-derived central genes. Moreover, sarcomere organization was found to be enriched. Indeed, sarcomere gene mutations are the main genetic cause of hypertrophic cardiomyopathy [366], which can lead to higher PP in patients with an abnormal blood pressure response [349]. Notch signaling pathway was a significant finding that independently replicated findings of altered Notch activity during TNFα-induced hypertension [367], as well as hypoxic pulmonary vasoconstriction [368]. All the aforementioned data may provide evidence of the biological utility of EpiDetect-derived epistatic networks in both replicating previous findings and potentially discovering novel significant pathways in complex phenotypes. The accuracy of EpiDetect might also warrant further investigation of the aforementioned novel pathways.

The main limitation of the approach lies in the scaling of the EpiCID component with respect to the number of SNPs analyzed. As the number of SNPs increases, the number of parameters that the network requires to learn increases, as well as the number of pairwise interaction strength computations to be done. Therefore,

this shows the importance of a prior filter on robust genome-wide significant signals to reduce the number of analyzed genetic variants. However, the computation time required for our studies, with the provided datasets described in Section 3.3.1, was more than reasonable. As an indicative measure for reference, on a machine with an Intel Core i7-12700H with 4.70 GHz of maximum clock speed, an NVIDIA RTX 3060 GPU with 6 GB of dedicated memory, and 16 GB of RAM, the training of the neural network took on average 30 minutes. The EpiCID explanation phase required less than 9 seconds (as average on SBP, DBP, and PP traits). At the time of writing, the work on EpiDetect was under consideration in a peer-reviewed journal.

Up to now, we have seen the first part of the explainable biomedical deep learning pipeline that concerns discovering disease-associated genes and determining their possible interactions. Once those genes have been found, they become part of the knowledge usable for further experiments and research. These can include additional disease gene prioritization studies, or the newly found genes can be leveraged for drug repurposing and discovery endeavors. We are thus advancing in the pipeline, going toward block 3 of Figure 1.1. As also described in Chapter 2, drug repurposing can be dealt with through both bioinformatics and chemoinformatics approaches. In this thesis, we will explicitly tackle drug repurposing from a bioinformatics perspective, employing NIAPU for the task, and this is why the work we are about to present finds its place within this chapter. However, the chemoinformatics methodologies that will be proposed in Chapter 4 for molecular activity and potency prediction are suitable for both drug repurposing and de novo design. The work we will present in the next section shows an effective application scenario for NIAPU as part of a drug repurposing pipeline. The same goal can also be achieved using XGDAG.

3.4 Network Proximity-Based Drug Repurposing for Primary Biliary Cholangitis

Upon the discovery of associated genes, it is possible to use them as targets for drug repurposing. This is what we did with primary biliary cholangitis (PBC). This disease lacks effective treatments, so finding new pathways and associated genes that can be used as drug targets is paramount. For those reasons, we chose PBC as a case study for the application of NIAPU as a means for drug repositioning. PBC is a chronic, cholestatic, immune-mediated, and progressive liver disorder that can also lead to malignant tumors [369], with a large percentage of transplants that are not successful [370]. Treatment to prevent the disease from advancing into later and irreversible stages is still an unmet clinical need. From a therapeutic standpoint, ursodeoxycholic acid (UDCA) is the first-line therapy for

all PBC patients and has been shown to slow the disease progression [371, 372]. However, only a small percentage of patients respond positively to the cure [373]. Obeticholic acid is the sole second-line therapy for patients who do not respond to UDCA [374, 373], and other drugs once deemed useful later proved to be ineffective and even dangerous [375, 376, 377, 378]. So, there is still room for research on novel possible treatments. Accordingly, we set up a drug repurposing framework to find potential therapeutic agents targeting relevant pathways derived from an expanded pool of genes involved in different stages of PBC. Starting with updated human PPI data and genes specifically involved in the early and late stages of PBC, NIAPU was used to provide a PBC gene ranking. When combined with already known PBC-associated genes, the top genes in the ranking resulted in a final set of genes most involved in the disease. Finally, a drug repurposing strategy was implemented by mining and utilizing dedicated drug–gene interaction and druggable genome information knowledge bases (e.g., the DrugBank [379, 380] repository). We identified several potential drug candidates interacting with PBC pathways using both known and NIAPU-detected genes. We found specific drugs as potential therapies targeting the distinct stages of the disease. The whole NIAPU-based pipeline is a robust and transparent selection mechanism for prioritizing already approved or investigational medicinal products for repurposing based on recognized unmet medical needs in PBC, helping identify a subset of drugs that could undergo clinical trials for specific usage with PBC in the future, in a safer and faster manner than the development of new medicines.

3.4.1 Methodology

This study was carried out via the following steps: i) literature search to gather known associated genes, ii) application of NIAPU to discover new candidate disease genes, and iii) drug repositioning using drug–target databases via enrichment analysis, backed up by pathway analysis of genetic information to determine the targeted pathways. We now report the details of each step.

3.4.1.1 Disease-Associated Gene Retrieval

A thorough search and filtering of the literature and databases were performed to compile a comprehensive genetic landscape of PBC. To gather gene–disease associations for PBC, both manual curation (relying on publications from MEDLINE and PubMed repositories) and automated retrieval (using DisGeNET [262, 48, 14]) were performed. The retrieved genes were also labeled by disease stage (early or late PBC) when clinically feasible according to studies or as *unspecified stages* (US) otherwise. The identification of features that characterize disease-associated genes, namely

genes experimentally associated with a specific disease, is critical for determining a complete genetic description of the pathology, assisting in the discovery of its etiology and potential treatments. Given a starting set of seed genes, the presence of characteristic patterns in their genetic, functional, or topological features can be used to better understand the disease's characteristics and to uncover new associated genes. A total of 1498 curated seed genes were found, which were then reduced to 1121 after data cleaning (duplicated removal after correcting gene symbol names to the HGNC human gene symbol standard [381]—a gene can have different names due to different naming conventions). These 1121 seed genes were then labeled according to their PBC stage: 238 early-stage genes, 183 late-stage genes, and 728 US genes, depending on the information provided by the specific article.

Each gene is assigned a relevance score that represents the degree of certainty that a seed gene is relevant to the disease. This score is assigned a value equal to the DisGeNET GDA score (also used in Section 3.1) for those genes present in the DisGeNET database. We remind the GDA score is a value ranging from 0 to 1, computed using the number and type of sources and the number of publications supporting the association [262]. With regard to the remaining manually curated genes, we assigned as score the maximum GDA score of the corresponding PBC disease stage (i.e., early stages, late stages, US). This choice was made in order to assign a higher weight to manually curated genes, which, thanks to the specific selection process, can be associated with the disease more reliably and robustly. Such genes have a major impact on the NIAPU network diffusion process (see Section 3.1.2). This manual curation answers the need for high-quality, more reliable, and better-curated GDA data [301], also discussed in Sections 3.1.5 and 3.2.2.1.

3.4.1.2 Network-Based Disease Gene Prioritization

Following the identification of the set of seed genes, we collected PPI data from BioGRID [13] as the first step to proceed with gene prioritization. As described also in Section 3.1, network medicine approaches exploit topological information deriving from the reconstruction and the analysis of PPI networks, as well as other features, to provide insights about the role of the gene in the onset and the development of the disease [259, 258]. Specifically, we applied NIAPU to provide a PBC-association gene ranking, using the manually curated and DisGeNET seed genes as the positive set P for the label propagation phase (stratified in three stage-related subsets). Consistent with the NIAPU system, we used the NeDBIT features to characterize the genes. Then, the label propagation phase yields a set of reliably putative disease-associated genes for any disease stage: the likely positive (LP) genes. The top 150 genes in the NIAPU ranking for each PBC stage were selected as LP genes and identified as

potential candidates for drug repurposing and pathway analysis, in addition to the original seed genes used as input.

3.4.1.3 Enrichment Analysis

WebGestalt (Web-based Gene Set Analysis Toolkit) [382] was used to perform functional enrichment analysis. *Homo sapiens* was chosen as the model organism, KEGG [383] and Reactome [384] were used as data sources for pathway enrichment analysis, and DrugBank [379, 380] and GLAD4U [385] resources were selected for drug repurposing gene–target analysis. Fisher's exact test-based over-representation enrichment analysis was conducted. Figure 3.15 depicts the analysis workflow from the initial curated genes (from the literature review and the DisGeNET database) to the LP genes obtained from the application of the NIAPU-based disease gene prioritization algorithm up to the drug repurposing and pathway analysis.

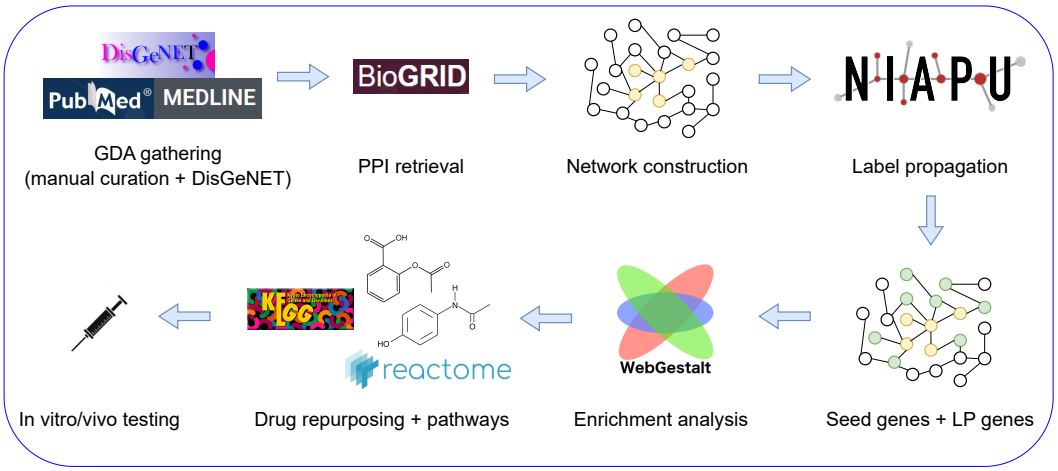


Figure 3.15. Workflow of the NIAPU-based drug repurposing pipeline. GDAs are gathered from DisGeNET and manually curated using PubMed and MEDLINE. PPI data from BioGRID are used to create a PBC-related network with the previously selected seed genes (in yellow). The network is fed to NIAPU for label propagation, which enlarges the set of associated genes with LP genes (in green). Enrichment analysis is performed with WebGestalt using seed and LP genes, yielding associated pathways and candidate drugs for future in vitro/vivo testing and clinical trials. The label propagation is repeated for every PBC stage.

3.4.2 Drug Repurposing Results

We will hereby present the results of the drug repurposing, considering separately the sets of already known seed genes and the LP set prioritized by NIAPU, stratifying the repurposed drugs by disease stage. For seed genes, the results of the most

significant drug–gene associations for each PBC stage are shown in Table 3.8, along with the number of genes responsible for repurposing the drugs.

Table 3.8. Drug repurposing results for seed genes. The most significant enriched drugs are reported, along with the number of genes by which they were repurposed and their p-value. The results are stratified by PBC stage.

Drug	PBC stage	Genes	p-value
Abciximab	Early	4	1.13e-5
Muromonab	Early	4	1.54e-5
Artenimol	Early	7	3.08e-5
Epipodophyllotoxin	Late	13	1.94e-10
TNF-α inhibitors	Late	14	2.26e-9
Interleukin inhibitors	US	74	$<$ 3.33e-16
Protein kinase inhibitors (PKIs)	US	61	$<$ 3.33e-16
Monoclonal antibodies	US	59	$<$ 3.33e-16
Specific immunoglobulins	US	57	$<$ 3.33e-16
Antineovascularisation agents	US	47	$<$ 3.33e-16
TNF-α inhibitors	US	47	$<$ 3.33e-16
Taurocholic acid (TUDCA)	US	27	3.77e-15
Anakinra	US	16	1.98e-13

Along with the drugs shown in Table 3.8, additional treatments were repurposed, regardless of the PBC stage. For instance, antivirals for systemic use and specific antirheumatic agents were significant findings (repurposed by 28 and 25 genes, respectively). Moreover, other significant and enriched results were etanercept and mycophenolate mofetil (13 and 9 genes). Furthermore, enzyme inhibitors (62 genes), drugs for musculoskeletal system disorders (28 genes), and corticosteroids, potent (group III) (6 genes) were repurposed. Additional results included antigout preparations (9 genes), biguanides (9 genes), simvastatin (14 genes), doxorubicin (20 genes), tamoxifen (12 genes), EGFR inhibitors (8 genes), and finally macrolides (13 genes). The same type of analysis was performed on the LP genes obtained using NIAPU, propagated from early-stage, late-stage, and US PBC seed genes. The most enriched drugs targeting LP genes are shown in Table 3.9.

The repurposed drugs have a clinical significance with validated therapeutic effects that will be discussed in Section 3.4.4.

Table 3.9. Drug repurposing results for LP genes. The most significant enriched drugs are reported, along with the number of genes by which they were repurposed and their p-value. The results are stratified by PBC stage.

Drug	PBC stage	Genes	p-value
L-lysine	Early	45	< 3.33e-16
L-threonine	Early	40	< 3.33e-16
Protein kinase inhibitors (PKIs)	Early	34	< 3.33e-16
Antineovascularisation agents	Early	22	9.35e-14
Enzyme inhibitors	Early	25	5.78e-11
L-serine	Early	35	1.98e-12
Erlotinib	Late	11	3.24e-14
EGFR inhibitors	Late	12	3.43e-13
Geldanamycin	Late	11	1.38e-12
Protein kinase inhibitors (PKIs)	US	53	< 3.33e-16
Geldanamycin	US	16	< 3.33e-16
Staurosporine	US	17	1.33e-15
Anti-estrogens	US	12	1.10e-09
Genistein	US	9	8.99e-10

3.4.3 Pathway Analysis Results

After having identified possible drugs targeting PBC-associated genes, including already known and newly found LP genes, pathway analysis was performed to back up the validity of the findings determining the pathways targeted by the repurposed drugs. In terms of the seed genes, the gene–pathway analysis accurately targeted a wide range of existing pathways, shown in Table 3.10.

Additional significant pathways discovered, regardless of the PBC stage, were measles (enriched by 34 genes), NF-κB signaling pathway (32 genes), Toll-like receptor (TLR) signaling pathway (31 genes), inflammatory bowel disease (IBD) (31 genes), and Th17 cell differentiation (30 genes). Moreover, for late-stage PBC, the IL-18 signaling pathway was significantly enriched by 20 genes. Likely, the same pathway analysis was performed on the LP genes propagated from the seed genes (Table 3.11).

In the following section, we will discuss the relevance of the pathways identified for PBC, along with the drugs by which they are targeted.

3.4.4 Observations

In this section, we have seen how to use NIAPU as the core of a drug repurposing pipeline. This strategy has several advantages over developing an entirely new

Table 3.10. Pathways found to be associated with seed genes. The most significant enriched pathways are reported, along with the number of genes involved in the enrichment and their p-value. The results are stratified by PBC stage.

Pathway	PBC stage	Genes	p-value
Innate immune system	Early	37	4.64e-12
Toll-like receptor cascade	Early	13	3.36e-9
Complement and coagulation cascades	Early	10	4.25e-9
Monoclonal antibodies	Late	26	< 3.33e-16
Epstein–Barr virus infection	Late	20	6.66e-16
Pathways in cancer	Late	29	1.22e-15
Immunoglobulins	Late	32	7.99e-15
Immune system	US	203	< 3.33e-16
Cytokine signaling in the immune system	US	114	< 3.33e-16
Signaling by interleukins	US	84	< 3.33e-16
Pathways in cancer	US	77	< 3.33e-16
Cytokine–cytokine receptor interaction	US	73	< 3.33e-16

Table 3.11. Pathways found to be associated with LP genes. The most significant enriched pathways are reported, along with the number of genes involved in the enrichment and their p-value. The results are stratified by PBC stage.

Pathway	PBC stage	Genes	p-value
Androgen receptor signaling pathway	Early	14	< 3.33e-16
Integrated breast cancer pathway	Early	18	3.88e-14
Ubiquitin-mediated proteolysis pathways	Early	17	6.84e-14
Complement and coagulation cascades	Late	15	9.78e-9
Allograft rejection pathway	Late	7	9.99e-5
Intestinal immune network for IgA production	Late	7	4.47e-4
Neurotrophin signaling pathway	US	19	3.33e-16
B-cell receptor signaling pathway	US	16	3.33e-16

drug for a specific indication. The risk of failure, for example, is reduced; because the repurposed drug has previously been demonstrated to be sufficiently safe in preclinical models and human trials, it is less likely to fail in later effectiveness trials, at least from a safety perspective. Furthermore, because the majority of preclinical testing, safety assessments, and, in some cases, formulation development will have already been completed, the time frame for drug development can be reduced. As we described, few cures are available for PBC, and those are not always effective. Based on these premises, establishing a drug repurposing framework to find potential therapeutic agents in PBC is highly relevant. We based our study on

the identification of drugs targeting relevant pathways derived from an expanded pool of genes by NIAPU involved in different stages of the disease, relying on a robust base of previously validated PBC disease genes. We reasoned that developing a drug repurposing framework would be extremely useful in identifying potential therapeutic agents in PBC, so we identified drugs targeting relevant pathways to aid in the understanding of the PBC molecular landscape, as well as the identification of genes that are not directly associated with it.

Our methodology reported both drugs previously evaluated with PBC and completely novel candidates. Concerning the interaction with seed genes, we identified several drug classes, including inhibitors of interleukin, protein kinase (PKIs), and TNF-α, as well as medications for musculoskeletal issues, TUDCA, immunosuppressants, antirheumatic agents, and simvastatin, as potential treatments for PBC, regardless of the disease stage. We also found that the immune system, cancer-related pathways, interleukin signaling, and cytokine receptor interactions, are closely linked to PBC. Additionally, pathways involving Th17 cell differentiation, TLRs, NF-κB signaling, and IBD were highly involved in PBC. NF-κB signaling is an important pathway because a substantial body of evidence suggests it plays a role in immunity, inflammation, cancer development, and nervous system function [386]. Indeed, in studies with mice, PBC has been shown to activate the NF-κB signaling pathway, leading to the release of inflammatory molecules and an increase in apoptotic proteins, resulting in liver injury [387]. Among interleukin inhibitors, ustekinumab, which is used to treat ulcerative colitis and Crohn's disease [388] was associated with modest benefits in PBC [389]. Anakinra, another repurposed drug, was effective in the treatment of patients with severe bacterial sepsis [390]. However, no clinical trials on PBC are currently underway. In a short-term trial, a PBC patient who didn't respond to UDCA (the standard treatment, as introduced) showed a significant response to the PKI baricitinib, despite potential side effects [391]. Moreover, our findings confirmed the already known efficacy of TUDCA in PBC [392, 393, 394]. Drugs that may modulate immunological abnormalities in PBC have been investigated, showing promise in slowing disease progression [395], confirming the immune system to be a relevant pathway in PBC. Lastly, drugs for musculoskeletal issues [396], particularly those targeting the RANK-RANKL axis, have shown potential with PBC, given their role in bile duct injury. In this, denosumab, a drug used to treat osteoporosis by inhibiting RANKL, might have the potential to preserve liver function in PBC, although it hasn't been investigated in this context.

From PBC early stages, the most enriched and significant pathways for seed genes were the innate immune system, TLR cascades, and complement and coagulation

cascades. For late stages, monoclonal antibodies, Epstein–Barr virus infection, cancer pathways, immunoglobulins, and the IL-18 signaling pathway were the most enriched and significant discoveries. Regarding the drugs repurposed, abciximab and muromonab were the most likely candidates for early PBC stages, whereas epipodophyllotoxin was a possibility for late stages. Abciximab, used in the past to reduce myocardial ischemic complications, may help due to its anti-inflammatory properties [397]. Muromonab blocks human T-cell functions and could benefit patients undergoing organ transplants [398, 399]. Epipodophyllotoxin derivatives, like etoposide and teniposide [400], are cancer drugs that inhibit specific enzymes with activity against drug-sensitive and drug-resistant cancer cells. Recently, etoposide has also been considered for treating cytokine storms in COVID-19 patients [401].

When we looked at the LP genes, we discovered different potential agents depending on the disease stage. For early stages, antineovascularization drugs, branched-chain amino acids (BCAAs), l-lysine and l-threonine, and enzyme inhibitors seemed promising. In late stages, drugs targeting the epidermal growth factor receptor (EGFR), such as erlotinib, were significant. Geldanamycin and staurosporine were potential treatments for unspecified stages of PBC. Notably, patients with PBC exhibit abnormal levels of BCAAs [402]. In particular, diminished levels of these amino acids (particularly l-phenylalanine and l-tyrosine) have been linked to chronic fatigue in PBC [403]. With regard to enzyme inhibitors, curcumin, a natural compound, has shown promise in addressing cholestasis, a condition where bile flow from the liver is impaired. It has also been found to play an antifibrosis role [404] and was recently evaluated for its safety and efficacy in patients with PSC (primary sclerosing cholangitis), whereas no previous or ongoing studies have evaluated its activity in PBC. Geldanamycin, an inhibitor of a protein called Hsp90, has shown potential in treating rheumatoid arthritis, a chronic inflammatory joint disorder (an autoimmune disease, like PBC). It works by specifically inhibiting the growth and inflammation of cells involved in rheumatoid arthritis. Geldanamycin has been studied in clinical trials for various types of blood and solid cancers and could be an interesting and promising candidate for PBC. Moreover, no data concerning the potential use for PBC were found with regard to staurosporine, a natural product with anticancer properties.

Another intriguing result is the enrichment found for anti-estrogens (tamoxifen may work similarly to UDCA [405], but there are no clinical trials testing anti-estrogen potential activity in PBC at the moment), along with androgen and integrated breast cancer signaling pathways for LP genes, regardless of their stages. In women and men, the immune system reacts differently. Adult females have higher innate

and adaptive immune responses than adult males. Women are more likely than men to develop autoimmune disorders, such as rheumatoid arthritis, multiple sclerosis, autoimmune liver diseases, and PBC, despite having a lower risk of developing most infectious diseases with a higher viral clearance [406]. The significance of estrogens in autoimmune illnesses has been thoroughly examined, and several lines of evidence and clinical observations indicate that sex hormones play a key role in disease etiology. Emerging proof suggests immunosuppressive effects of androgens [407]. The discovery of alterations in testosterone serum levels in mice connected to the intestinal microbiota should pique interest in the function of the microbiome in sex differences in autoimmune liver disorders, which are linked to an altered intestinal microbiota. The intestinal immune network for IgA production signaling and IBD pathways were found to be enriched for LP genes for PBC late stages. The most striking finding is the high prevalence of IgA anti-calreticulin antibodies and its class pattern in PBC patients, suggesting a reactivity of the gut-associated immune system, which could imply that a yet-to-be-identified gut-derived bacterial agent could be a potential actor in the onset of PBC. Also related to sex-dependent immune response, genistein, a bioflavonoid, was also found to be enriched from LP genes, regardless of their disease stage. Among its mechanisms, genistein has shown a growth-inhibitory effect on human cholangiocarcinoma cells. Despite such compounds being found to decrease liver fibrosis and cholestasis in rats [408], genistein has not yet been studied in a clinical trial setting.

In summary, using NIAPU, we conducted a study on a large dataset of PBC-curated genes from credible and publicly available sources obtaining a list of new potential disease genes. These genes were then enriched for biological pathways and drugs to obtain new potential insights for PBC pathogenesis and treatment, proposing potential drug candidates for distinct stages of the disease. We identified novel therapeutic targets for prioritization in PBC and new pathogenic pathways using a framework that provides a better definition of the PBC molecular landscape. We provided a robust and transparent selection mechanism for prioritizing already approved medicinal or investigational products for repurposing based on recognized unmet medical needs in PBC and sound preliminary data in order to identify research priorities for a better understanding of the mechanisms of action of drug candidates via future ad hoc, in vitro, and in vivo tests and clinical trials.

The identification of multiple non-specific liver pathways may shed new light on the extrahepatic pathogenesis of PBC, where gut microbiota, sex hormone-receptor interactions, and bone marrow interplay may all play a role, to varying degrees, at different stages of the disease. In the first phase, BCAAs, geldanamycin, taurour-

sodeoxycholic acid, bioflavonoids (particularly genistein), anti-estrogens, curcumin, monoclonal antibodies, antineovascularisation, and antirheumatic agents are the most interesting therapeutic candidates worthy of evaluation in PBC experiments. Moreover, pharmacological categories such as specific interleukin/EGFR/TNF-α inhibitors could be tested in particularly advanced disease stages. The results of our drug repurposing studies were published in *Biomedicines* [24].

The successful application of NIAPU as a means for prioritizing genes for drug repurposing allowed us to present an example of a bioinformatics-driven drug repurposing framework. As we anticipated, the same pipeline can feature XGDAG in lieu of NIAPU for disease gene prioritization to leverage the power of graph neural networks and XAI for drug repositioning in the future.

With this work, we concluded the description of the bioinformatics components of the explainable biomedical deep learning pipeline. In the next chapter, we will dive into the chemoinformatics part, starting with a GNN-based strategy for molecular activity prediction, usable in drug development for de novo design or repurposing.

Chapter 4

Chemoinformatics and Medicinal Chemistry

In the previous chapter, we presented the bioinformatics part of the pipeline. This chapter is dedicated to the second and last macro-area: chemoinformatics. We will now focus on molecular activity prediction and protein–ligand interaction, pillar tasks in drug discovery. The methods proposed will fall in block 3 of the explainable biomedical deep learning pipeline in Figure 1.1, which can be considered as a bridge between bioinformatics and chemoinformatics, and in block 4, which covers the explainability solutions for deep learning methods for drugs development. We will start by presenting EdgeSHAPer, our strategy to explain graph neural networks (GNNs) using approximated Shapley values in the context of compound activity prediction (Section 4.1), and then we will use it to uncover what GNNs really learn when applied to the task of potency prediction in protein–ligand interactions (Section 4.2). Finally, in Section 4.3, we will take a step back from neural networks in favor of simpler support vector machine (SVM) models; this will allow us to devise an innovative explainable artificial intelligence (XAI) methodology based on exact Shapley value computation rather than on their approximation, solving the issues related to the latter.

4.1 Shapley Value-Based Explanation Method for Graph Neural Networks in Molecular Activity Prediction

As anticipated, a central task in drug design is the prediction of molecular activity. Once a possible drug target for a disease is identified, for instance, using proposed techniques like the NIAPU or XGDAG, it is time to look for a candidate compound that can be the starting point of the drug to be developed. This molecular compound has to be active on the target and be able to bind and modulate it. Classification

models can be trained to predict if a given molecule is active against the target of interest. Virtual screening can be used to find active compounds. As we described in Section 2.2, many successful machine and deep learning applications have been employed, such as SVMs, random forests (RFs), and deep learning strategies, from simple mutlilayer perceptrons (MLPs) to more complex convolutional (CNNs) and recurrent neural networks (RNNs) [100]. These methodologies rely on precomputed features or grid-like representations obtained from molecular structures and atomic information. Along with these models, GNNs represent an increasingly popular class of neural networks for deep learning in drug design, with message-passing neural networks (MPNNs) being a prominent example [20, 119]. This is partly due to their ability to learn directly from graph representations, which alleviates the need for predefined features and descriptor engineering. These GNNs are particularly attractive for representation learning in chemistry [119], given that molecular graphs are the primary data structure for conveying molecular information, implicit structure-based properties, or molecular interactions. In a typical molecular graph, nodes represent atoms and edges represent bonds between atoms. For our study, we decided to use GNNs to explicitly exploit the molecule structure and capture the knowledge embedded in molecular graphs. Like other neural networks, GNNs have a black-box character, which also confines their acceptance in chemistry. This is why we now report the development and assessment of a new explanation method for GNNs, which we called EdgeSHAPer, able to quantify edge importance for GNN predictions using an effective Shapley value approximation based on Monte Carlo sampling.

4.1.1 Scientific Context

The EdgeSHAPer approach introduced herein was devised to assess edge importance for GNN predictions using Shapley values. As described in the related work in Section 2.3.1, in literature, we find few XAI methods for GNNs that focus their explanations on edge importance. For instance, GNNExplainer, the first GNN explanation method developed, learns a mask on node features and on the adjacency matrix applied to identify the subgraph for an object determining its prediction [225]. In addition to GNNExplainer, PGExplainer was introduced as a parameterized strategy suitable for inductive settings [226]. However, there are only a few more approaches currently available to aid in rationalizing GNN learning, as introduced and also further discussed below, which employ the Shapley value concept.

EdgeSHAPer was originally conceptualized for assessing the importance of bond information for graph-based compound activity prediction, representing a novel

approach, and was specifically evaluated in this context. Compound activity prediction is a central task for machine learning in chemoinformatics and medicinal chemistry. Bonds between atoms are key elements in a molecule, and determining edge importance in molecular graphs can help identify substructures responsible for compound activity. However, our new methodology is generalizable and applicable to many tasks in GNN learning where edge distribution plays a role, including any node degree-sensitive MPNNs.

4.1.1.1 Shapley Values in Explainable Machine Learning

EdgeSHAPer makes use of Shapley values, introduced in game theory [26] to quantify the contributions of individual players to the performance of a team. As described in Section 2.3.2, the Shapley value concept was adapted to be used in XAI as a model-agnostic framework to rationalize predictions of machine learning models. In this context, Shapley values are calculated to quantitatively assess feature importance for individual predictions. Since the calculation of Shapley values depends on the order of players (features) and is thus combinatorial in nature, it becomes computationally demanding in high-dimensional feature spaces (typically used in chemoinformatics applications). Therefore, the Shapley additive explanations (SHAP) approach has been introduced, which approximates a machine learning model in the feature space vicinity of a test instance relying on a local model based on a kernel function [8]. SHAP can be perceived as Shapley value-based extension of the Local Interpretable Model-agnostic Explanations (LIME) approach [222]. SHAP-based methodologies have also been introduced and evaluated for compound activity, multi-target activity, and potency predictions [239, 204]. While SHAP-based explanations have been proposed for rationalizing different types of activity predictions in chemoinformatics, they have exclusively been applied to machine and deep learning models trained using precomputed descriptors [129], but never on graph representations.

The Shapley value concept has recently been applied to graphs in other fields. We remind GraphSVX [240] was introduced as a decomposition method for GNNs that relies on a linear approximation of Shapley values to determine node and node feature contributions. In addition, SubgraphX [241] was developed as a subgraph-centric method. It approximates Shapley values to find the most critical fully connected subgraph for the prediction, being the first methodology to consider connected graphs as explanations. Finally, GRAPHSHAP [242] was devised as a motif-focused XAI approach for graph classification with node awareness [243].

While these SHAP-based methodologies produce explanations focused on nodes,

subgraphs, or motifs, none of them quantifies edge importance, although graph information is primarily distributed through edges. The missing SHAP-dependent quantification of edge importance for GNN predictions has partly motivated the development of our new approach in the context of molecular graphs. The principal methodological differences between EdgeSHAPer, as introduced herein, and the other SHAP-based explanatory approaches for graph learning preclude a meaningful direct comparison. However, in light of edge centricity, the results of EdgeSHAPer applications can be compared to those of GNNExplainer, although the approaches are also conceptually distinct. Moreover, we compare EdgeSHAPer explanations against another Shapley value-based strategy, the TreeExplainer variant of SHAP applied to a random forest model.

4.1.2 The Algorithm

The Shapley value concept has been adapted for EdgeSHAPer using the following analogies: we consider a setting in which players corresponding to edges in a graph work collaboratively toward a team (graph) reward, which represents the probability of a prediction for a test instance obtained with a machine learning model. Each player makes an individual contribution to the reward (payout), which is represented by its Shapley value and computed as the average marginal contribution over all possible feature coalitions (orderings). Since enumerating all possible coalitions becomes computationally hard for larger feature sets, Shapley values are approximated for machine learning applications.

In our approach, each edge of a graph has its own payout contribution to the predicted output probability (value v). Adapting the Shapely value definition from Equation (2.1) in Section 2.3.2 using edges as features/players, the Shapley value for edge j is computed as:

$$\phi_j(v) = \frac{1}{|E|} \sum_{S \subseteq E \setminus \{j\}} \frac{v\left(S \cup \{j\}\right) - v(S)}{\binom{|E|-1}{|S|}}, \tag{4.1}$$

where E is the set of all edges and $|E|$ its cardinality, S indicates all the possible subsets of edges excluding j and $|S|$ its cardinality, $v(S)$ is the value achieved by subset S, and $v(S \cup j)$ is the value obtained when edge j joins the subset S (considering the edge's marginal contribution).

4.1.2.1 Monte Carlo Sampling of Edges

In machine learning, the practical inability to compute Shapley values directly in many cases requires the use of approximation methods. We developed a Monte Carlo sampling strategy for graph edges, which is central to the EdgeSHAPer algorithm. Instead of randomly sampling a data point from a dataset [236], which is not applicable in this context, we generate a random graph Z that contains the same number of nodes as the explained graph G according to a binomial probability distribution. If an edge e exists in G, it exists in Z with a probability equal to some P. The density of graph G, which is analogous to the probability for an edge to exist in this graph, proved to be a meaningful choice for P, as further described below. At any Monte Carlo step, a new graph Z is generated. The complete pseudocode for the EdgeSHAPer algorithm with Monte Carlo sampling is provided in Algorithm 1.

Algorithm 1 EdgeSHAPer with Monte Carlo sampling

Require: $G(N, E), j, P, M, \hat{f}$

$cumulative_{\phi_j}(G) \leftarrow 0$
for each $m \in \{0, \ldots, M-1\}$ **do**
  $N_z \leftarrow N$
  $E_z^{mask} \leftarrow binomial(P, |E|)$
  $\pi \leftarrow permutation(|E|)$
  $j^\pi \leftarrow \pi(j)$
  $E^{mask} \leftarrow list(1, |E|)$
  $E^\pi \leftarrow \pi(E^{mask})$
  $E_z^\pi \leftarrow \pi(E_z^{mask})$
  $E_{+j}^{mask} \leftarrow (e_0^\pi, \ldots, e_{j^\pi}^\pi, z_{j^\pi+1}^\pi, \ldots, z_{|E|-1}^\pi)$
  $E_{-j}^{mask} \leftarrow (e_0^\pi, \ldots, e_{j^\pi-1}^\pi, z_{j^\pi}^\pi, z_{j^\pi+1}^\pi, \ldots, z_{|E|-1}^\pi)$
  $E_{+j} \leftarrow select\_from\_mask(\pi(E), E_{+j}^{mask})$
  $E_{-j} \leftarrow select\_from\_mask(\pi(E), E_{-j}^{mask})$
  $\phi_j^m(G) \leftarrow \hat{f}(N_z, E_{+j}) - \hat{f}(N_z, E_{-j})$
  $cumulative_{\phi_j}(G) \leftarrow cumulative_{\phi_j}(G) + \phi_j^m(G)$
end for
$\phi_j(G) \leftarrow cumulative_{\phi_j}(G)/M$
return $\phi_j(G)$

Here, G is the graph to explain, E the list of edges of this graph, and N are the nodes; j is the edge for which the current Shapley value is computed, P the probability of an edge from E to exist in graph Z (density of G in our implementation), M the number of Monte Carlo steps (corresponding to the number of randomly generated graphs

Z), and $\hat{f}$ is the function learned by the GNN. E_z^{mask} is a binary mask indicating if an edge from E is present in graph Z. As we can notice, there is no need to explicitly define Z in the algorithm since we will only need its edges for the subsequent steps. Hence, EdgeSHAPer creates a random permutation π and sorts the edges of G and Z according to this permutation creating permuted masks defining the presence or absence of edges (E^{π} and E_z^{π}). Then, still relying on binary masks, two edge lists are created by appending edges from the two permuted lists, considering the permuted position of j, j^{π}, as a split point: in E_{+j} edge j originates from the original graph G, while in E_{-j} its counterpart originates from Z. Thereby, the contribution of an edge to the output is calculated. The algorithm is repeated for each edge in the graph.

It is possible to express the EdgeSHAPer algorithm in a more compact and concise way, as shown in Algorithm 2. However, the pseudocode proposed in Algorithm 1, with the usage of binary masks, resembles the actual implementation more closely.

Algorithm 2 EdgeSHAPer with Monte Carlo sampling - Alternative

Require: $G(N, E), j, P, M, \hat{f}$

$cumulative_{\phi_j}(G) \leftarrow 0$
for each $m \in \{0, \ldots, M-1\}$ **do**
  $N_z \leftarrow N$
  $E_z^{mask} \leftarrow binomial(P, |E|)$
  $E_z \leftarrow select\_from\_mask(E, E_z^{mask})$
  $\pi \leftarrow permutation(|E|)$
  $E_{+j} = \{e : e \in E \wedge \pi(e) \leq \pi(j)\} \cup \{e : e \in E_z \wedge \pi(e) > \pi(j)\}$
  $E_{-j} = \{e : e \in E \wedge \pi(e) < \pi(j)\} \cup \{e : e \in E_z \wedge \pi(e) \geq \pi(j)\}$
  $\phi_j^m(G) \leftarrow \hat{f}(N_z, E_{+j}) - \hat{f}(N_z, E_{-j})$
  $cumulative_{\phi_j}(G) \leftarrow cumulative_{\phi_j}(G) + \phi_j^m(G)$
end for
$\phi_j(G) \leftarrow cumulative_{\phi_j}(G)/M$
return $\phi_j(G)$

Notably, the random graph used for Monte Carlo sampling is not an Erdős–Rényi random graph [409]. Here, an edge exists with probability P in the generated random graph only if it also exists in the original molecular graph. This enables the quantification of specific edge contributions in coalitions with other edges and, thus, the determination of the importance of a particular bond in a given compound. The underlying idea is the use of random graphs starting from a test molecule to define information baselines relative to which the contribution of each edge/bond can be

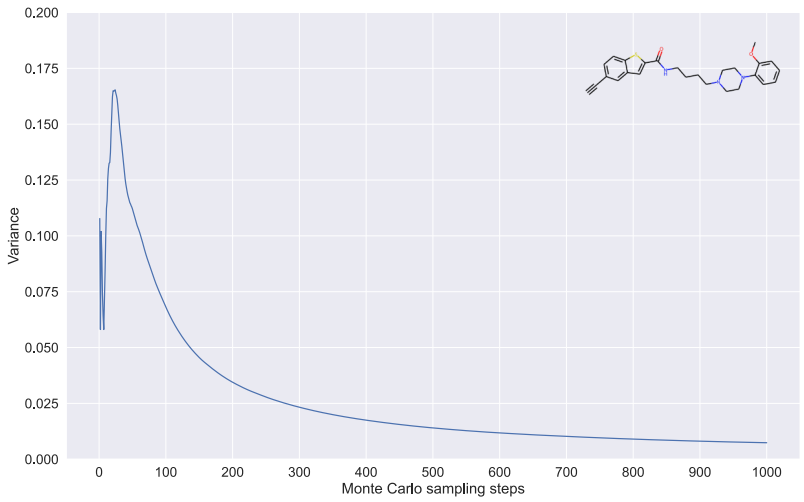
quantified. Moreover, the use of this specifically generated random graph enables the generalization of EdgeSHAPer for applications in different domains.

Consistent with GNN learning, EdgeSHAPer considers both directions for edges. This is in line with how information is diffused in GNNs. When a graph is undirected, information flows in both directions; hence, each direction will have its own contribution to the model's prediction (of course, in the case of directed graphs, only one direction is considered). Thus, given the additivity property of Shapley values [26] (Section 2.3.2), the total contribution of an edge can be calculated by summing the Shapley values for the two directions. The final output of the algorithm is a ranking of edges on the basis of approximated Shapley values.

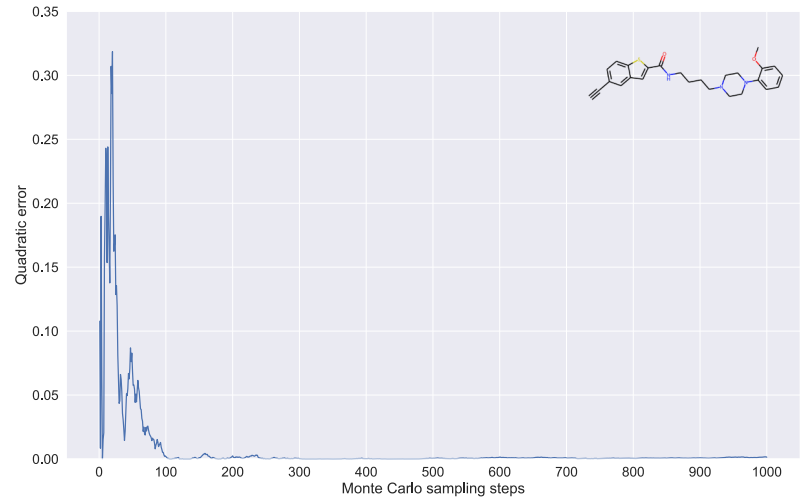
To study the evolution of the approximation over sampling steps and determine the number of steps required for a reliable approximation value, we analyzed the variance and convergence for EdgeSHAPer. The random variable was given by the sum of Shapley values $\phi_j(G)$ for any edge j of the graph and the expected value by the difference between the output probability and the average prediction. In fact, as per the efficiency property [26] (see Section 2.3.2), the sum of Shapley values corresponds to the deviation of the actual prediction from the average. In our specific case, however, the average prediction is to be intended on the randomly generated graphs Z. Since we are interested in the contributions to the prediction of active compounds, the average prediction for the random graphs to be active is close to zero. Consequently, the sum of Shapley values corresponds to the output probability in this case. Figure 4.1a shows the evolution of the variance, and Figure 4.1b shows the quadratic error for a test compound representing the deviation between the predicted probability and the sum of Shapley values.

Increasing numbers of Monte Carlo sampling steps yielded an accurate and stable approximation of the prediction probability as the sum of the Shapley values. During sampling, the variance decreased asymptotically against 0 (Figure 4.1a), and the quadratic error was already very close to 0 after only 100 steps (Figure 4.1b). Therefore, given the need to evaluate EdgeSHAPer on a large set of samples, $M = 100$ was considered a proper choice, providing a favorable compromise between approximation accuracy and computational time requirements. In addition, a termination criterion was implemented for the sampling procedure, defining a permitted deviation between the sum of the Shapley values and the predicted probability.

Specific evaluation metrics To quantitatively evaluate the performance of GNN explanation methods, two metrics were introduced including FID+ (Fidelity) and



(a) Variance of Shapley values.



(b) Error convergence.

Figure 4.1. EdgeSHAPer variance and error convergence. Shown are variance **(a)** and error convergence **(b)** for an exemplary test compound over increasing numbers of Monte Carlo sampling steps.

FID− (Infidelity) [410]. These metrics evaluate the quality of unimportant and important features, respectively, and are defined as

$$\text{FID+} = \frac{1}{n}\sum_{i=0}^{n-1}\left(f(G_i) - f(U_i)\right)$$

and

$$\text{FID}- = \frac{1}{n}\sum_{i=0}^{n-1}\left(f(G_i) - f(I_i)\right),$$

where G_i is the original graph, U_i is the graph obtained from G_i exclusively containing unimportant features (nodes, edges, or node/edge features), I_i is the graph obtained by G_i exclusively containing important features, and n is the number of samples (graphs) for which the metric is computed. Herein, the probability version of FID+ and FID− was used, as introduced previously [410]. A deep learning model with meaningful feature representation should tend to produce high FID+ and low FID− scores.

In our work, we used an adapted version of these metrics relying on minimal sets of relevant features. The pertinent positive set (P_{POS}) [411, 412] represents the minimal set of features required for a given class label prediction of an instance. Moreover, we defined the minimal top-k set (T_k) as the minimal set of features that must be removed to invert the class label (here from active to inactive). Those sets comprise the features with the highest Shapley value estimates from EdgeSHAPer. P_{POS} is created in an inductive manner by adding edges with the highest Shapley values one by one to the graph until a test compound is correctly predicted to be active. By contrast, T_k is obtained following a deductive approach; starting from a compound correctly predicted to be active, the most important edges are removed until the molecule is classified as inactive. The consideration of such minimal feature sets determining class label predictions is related to the concept of contrastive explanations [413, 208]. This feature selection scheme ensured that the most influential features for predictions were identified on the basis of (molecular) graphs with varying numbers of edges (bonds). FID+ and FID− are computed using T_k and P_{POS}, respectively.

4.1.3 Compound Classification

To provide a meaningful basis for the assessment and comparison of explanation methods, we selected a test case that was expected to yield high classification

accuracy based on prior experience. Therefore, compounds with activity against the dopamine D2 receptor were selected. Compounds and corresponding exact standard potency measurements (Ki, Kd, or IC50) of at least 10 μM were obtained from ChEMBL (version 29) [414, 415, 416]. Those are measures used to evaluate, with different criteria, the binding affinity of a compound with a target. Ki is the inhibition constant, which measures the reduction in the activity of the target; Kd is the dissociation constant and measures the equilibrium between the ligand–protein complex and the dissociated components; IC50 stands for inhibitory concentration 50%, meaning the concentration of inhibitor needed to halve the biological activity of the target. Only direct interactions against human wild-type proteins at the highest target confidence level were retained. Using publicly available filters [417, 418, 419], molecules exceeding a mass of 1000 Da were removed along with potential assay interference compounds. Based on this protocol, 4174 active compounds were obtained and complemented with an equal number of randomly selected compounds (omitting ligands with activity against functionally related G protein-coupled receptors). The compound dataset was divided into training (80%), validation (10%), and test (10%) sets.

Graph convolutional network model Any GNN model can be explained using EdgeSHAPer. For our proof-of-concept study, we used a graph convolutional network (GCN) [186] due to its increasing popularity in chemistry [420]. The model was constituted of four convolutional layers with 256 hidden units and a rectified linear unit (ReLU) as an activation function to introduce nonlinearity. Global mean pooling and dropout with a probability of 0.5 were considered. The GCN was trained for 100 epochs with a batch size of 32, Adam optimizer [307], and a learning rate of 0.001. The model was implemented in PyTorch [421] using the PyTorch Geometric library [304]. The GCN operator is defined as

$$\mathbf{X}' = \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{\Theta}, \tag{4.2}$$

where $\hat{\mathbf{A}}$ is the adjacency matrix considering self-loops ($\hat{\mathbf{A}} = \mathbf{A}+\mathbf{I}$), $\hat{\mathbf{D}}$ is the diagonal degree matrix of $\hat{\mathbf{A}}$ ($\widehat{D}_{ii} = \sum_{j=0} \widehat{A}_{ij}$), and $\mathbf{\Theta}$ are the neural network weights.

Random forest classifier The RF algorithm [107] consists of an ensemble of decision trees built with bootstrapping and feature bagging. The scikit-learn RF implementation was utilized [422]. For RF classification, structural features of compounds were generated and hashed using the RDKit implementation of the

Morgan fingerprint with a bond radius of 2 [423, 419]. The presence or absence of generated features was recorded in a binary feature vector in which features were mapped to unique positions. RF classifiers with different hyperparameter settings were derived. The grid search included hyperparameters as the number of decision trees, the minimum number of samples per node split, and the minimum number of samples per leaf node. Hyperparameter value combinations with the highest balanced accuracy over ten independent training/validation partitions were used to derive the final classifier on the complete training set. Exact Shapley values for comparison for predicted class probabilities of RF classifier (fraction of positive predictions in the tree ensemble) were calculated using the TreeExplainer algorithm with the interventional feature perturbation approach, for which the training data served as a background sample [8, 239].

We applied GCN and RF models to a compound classification task aiming to systematically distinguish between dopamine D2 receptor ligands and other randomly selected compounds. A balanced accuracy of 0.99 for the RF model was obtained for the test set. Furthermore, 99% of the active compounds were successfully identified while maintaining a high precision of 0.99. The GCN model also achieved a high balanced accuracy of 0.97 for the test set. In addition, to evaluate the stability of the predictive performance and model explanations, the training set was divided into three disjoint subsets, and the GCN was re-trained on each of these size-reduced partitions. Despite the smaller number of training samples, only a slightly lower mean classification balanced accuracy of 0.95 was obtained. Hence, these results confirmed the stability of the GCN predictions. The high level of classification accuracy achieved by RF and alternative GCN models provided a sound basis for explaining compound activity predictions and comparing different methods. Predictions of these models were first used to evaluate the consistency of EdgeSHAPer explanations, followed by orthogonal feature mapping analysis in comparison to TreeExplainer for RF as well as quantitative and qualitative comparisons to GNNExplainer.

4.1.4 Explaining Graph Convolutional Network Predictions

We will now analyze the explanations returned by EdgeSHAPer. Firstly, we will investigate their robustness and consistency; then, we will compare them against GNNExplainer and TreeExplainer outputs to evaluate their accuracy.

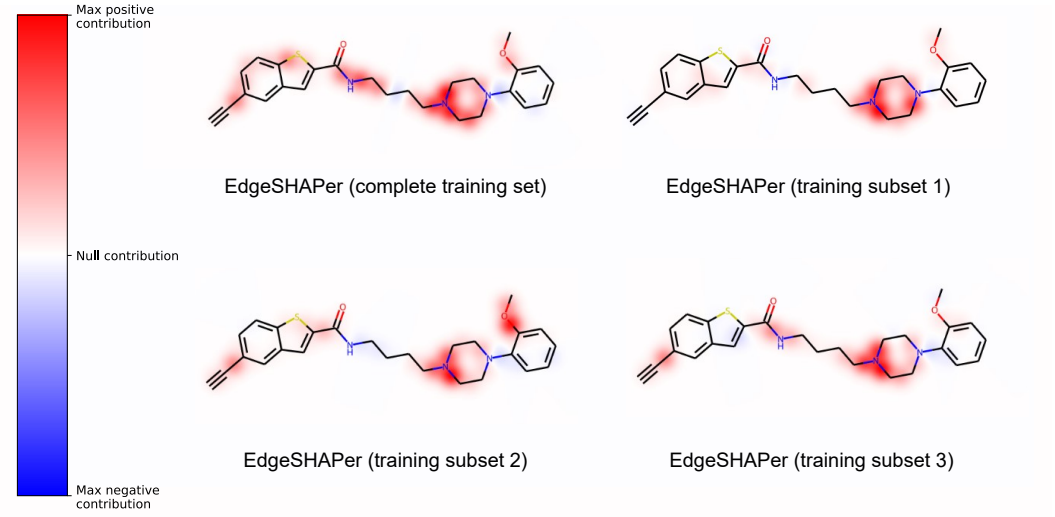
4.1.4.1 Consistency of the Explanations

Initially, we evaluated EdgeSHAPer explanations and their consistency for training sets of different sizes and compositions. EdgeSHAPer was applied to multiple

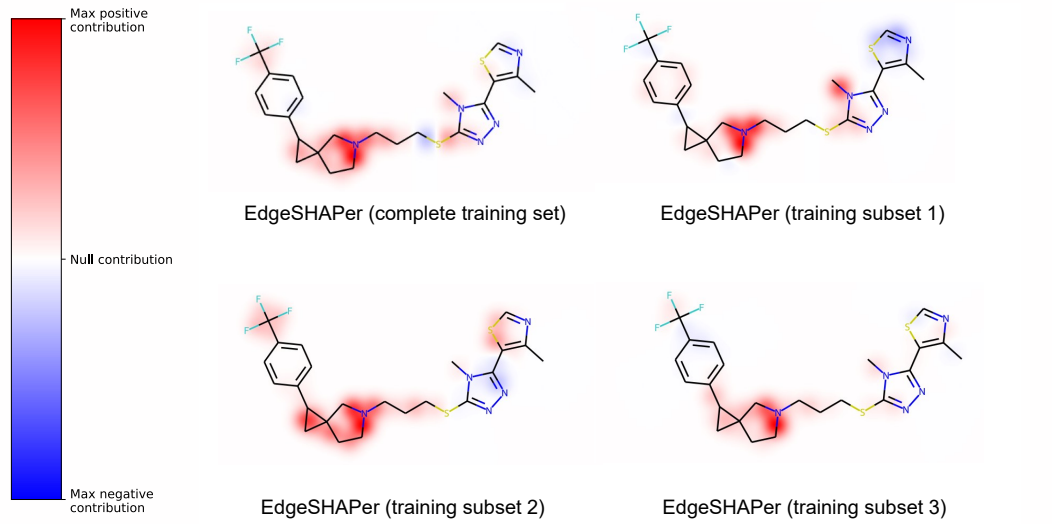
GCN models derived on the basis of a complete training set or random training data subsets. These explanation results were quantitatively and qualitatively compared. Quantitative comparisons were carried out on the basis of the FID+ and FID− metric variants to assess minimal feature sets determining correct predictions of active compounds. Qualitative comparison with feature visualizations was also obtained by mapping minimal feature sets on correctly predicted test compounds.

Edges prioritized by EdgeSHAPer were mapped on test compounds (Figure 4.2). In this and the following figures, coloring identifies the most important edges representing covalent bonds. Red coloring indicates positive (supporting the prediction) and blue negative contributions (opposing the prediction)—the intensity of the color scales with increasing edge importance. For test compounds belonging to different chemical series, depicted in Figures 4.2a and 4.2b, respectively, feature mapping revealed that edges prioritized by EdgeSHAPer consistently formed the same coherent substructures in test compounds predicted with GCN models derived on full and partial training sets. Minor differences between features prioritized using non-overlapping subsets with distinct compounds are expected. Importantly, for each chemical series, the same coherent substructures responsible for correct predictions were identified in different test compounds using distinct subsets of only one-third of the size of the original training set, indicating the stability of the EdgeSHAPer results. For GCN models generated with different training subsets of reduced size, the identified substructures were slightly smaller than for the model trained on the complete training set due to the lower number of training instances and features in subsets. It is emphasized that the formation of coherent substructures of limited size by prioritized features in both compound series revealed that these features delineated chemically meaningful substructures determining the predictions. Furthermore, as also shown in Figure 4.2, positive contributions clearly dominated correct compound activity predictions, with only very little balancing influence of negative contributions.

Visual analysis was complemented and confirmed by the quantitative assessment in Table 4.1, reporting differences based on FID+ and FID− values and the cardinalities of the minimally informative sets. Hence, EdgeSHAPer explanations were non-ambiguous, consistent, and stable.



(a) Compound C#Cc1ccc2sc(C(=O)NCCCCN3CCN(c4ccccc4OC)CC3)cc2c1



(b) Compound Cc1ncsc1-c1nnc(SCCCN2CCC3(CC3c3ccc(C(F)(F)F)cc3)C2)n1C

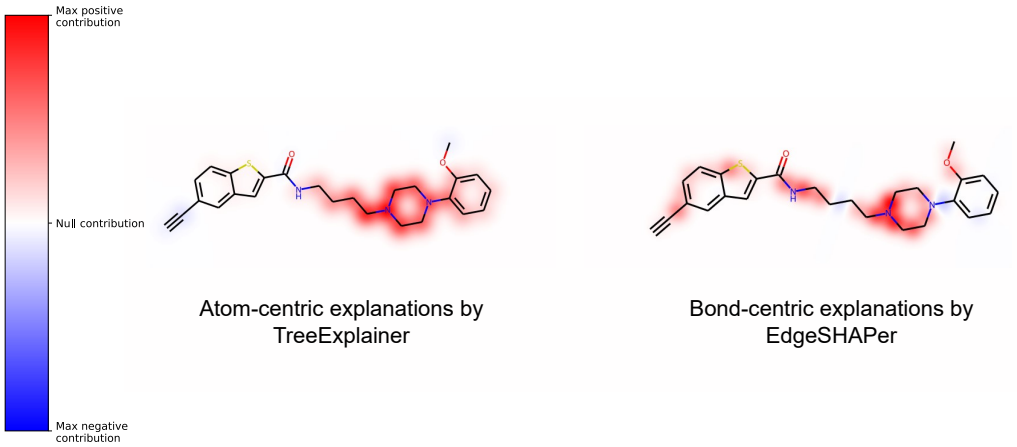
Figure 4.2. In **(a)** and **(b)**, explanations are provided for exemplary test compounds.

Table 4.1. Mean test set FID+ and FID− scores for EdgeSHAPer and the complete training set as well as non-overlapping subsets of the training set and the mean number of edges comprising the minimal sets.

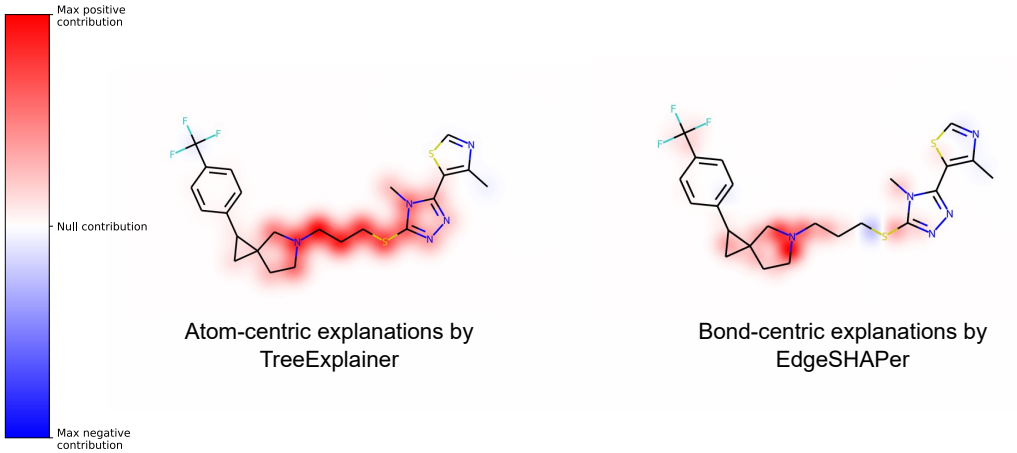
	FID+	**FID−**	**Edges in P_{POS}**	**Edges in T_k**
Training set	0.934	0.137	12.85	3.75
Training subset 1	0.886	0.108	5.50	4.80
Training subset 2	0.851	0.245	7.20	3.75
Training subset 3	0.926	0.120	7.45	4.10

4.1.4.2 Comparison with TreeExplainer

An orthogonal qualitative comparison of features determining GCN and RF predictions was also carried out. Therefore, the EdgeSHAPer and TreeExplainer methods were applied to rationalize GCN and RF predictions, respectively. In this case, substructures delineated by principally distinct molecular features, that is, predefined structural features for RF and the representation learned by GCN, were compared. For this analysis, RF models were implemented in combination with TreeExplainer since it enables exact (rather than locally approximated) calculation of Shapley values for decision tree methods and is node (atom)-centric, in contrast to EdgeSHAPer, which is edge (bond)-centric. Figure 4.3 shows representative results.



(a) Compound C#Cc1ccc2sc(C(=O)NCCCCN3CCN(c4ccccc4OC)CC3)cc2c1



(b) Compound Cc1ncsc1-c1nnc(SCCCN2CCC3(CC3c3ccc(C(F)(F)F)cc3)C2)n1C

Figure 4.3. In **(a)** and **(b)**, mappings are shown for exemplary test compounds comparing explanations from EdgeSHAPer and TreeExplainer.

Even if applied to different machine learning algorithms relying on learned representation features against predefined descriptors (GCN and RF, respectively), EdgeSHAPer bond-centric and TreeExplainer atom-centric explanations delineated overlapping yet distinct substructures responsible for correct predictions. While these results were not necessarily expected, they supported the relevance and robustness of the SHAP/Shapley value-based explanatory framework. Notably, substructures identified by EdgeSHAPer explanations were smaller than those found by TreeExplainer, which either resulted from the different features used or corresponded to the higher resolution of EdgeSHAPer explanations, focusing on substructures decisive for predictions.

4.1.4.3 Comparison with GNNExplainer

EdgeSHAPer was then compared to GNNExplainer, which also exclusively considers edges for model explanation and does not employ other local approximation methods. The same quantitative/qualitative analysis scheme as above was applied. Table 4.2 reports the quantitative comparison. EdgeSHAPer identified smaller pertinent positive sets of chemical bonds required for accurate predictions, similar to the abovementioned observations. Furthermore, EdgeSHAPer yielded higher FID+ scores than GNNExplainer and identified smaller minimal top-k sets. GNNExplainer produced low FID− scores since it identified minimal sets with larger numbers of edges. Indeed, pertinent positive sets with increasing numbers of features rendered predicted probabilities close to the original probability of a prediction, which led to decreasing FID− values. However, EdgeSHAPer scores were of lower magnitude, showing that its smaller pertinent positive sets conveyed important information. Table 4.3 shows the comparison of the explanations for the training subsets, again confirming the stability of the results and higher resolution of the EdgeSHAPer explanations.

Table 4.2. Mean test set FID+ and FID− scores for EdgeSHAPer and GNNExplainer for the complete training set and mean number of edges comprising the minimal sets.

	FID+	**FID−**	**Edges in P_{POS}**	**Edges in T_k**
EdgeSHAPer	0.934	0.137	12.85	3.75
GNNExplainer	0.813	0.154	31.10	15.55

Feature mapping gave consistent results (Figure 4.4). As observed in the comparisons discussed above, EdgeSHAPer identified small coherent substructures in test compounds driving correct predictions, whereas features prioritized by GNNExplainer

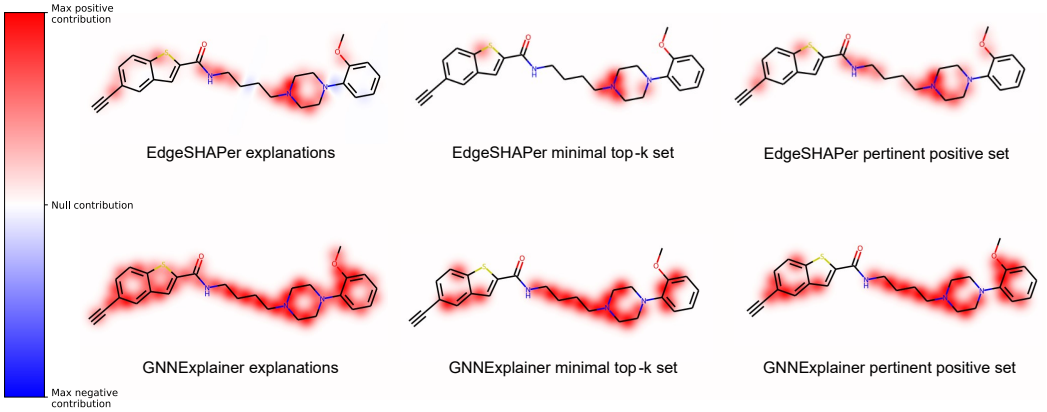
Table 4.3. Mean test set FID+ and FID− scores for EdgeSHAPer and GNNExplainer for the training subsets and mean number of edges comprising the minimal sets of the subsets.

	FID+	**FID−**	**Edges in P_{POS}**	**Edges in T_k**
EdgeSHAPer	0.888	0.158	6.72	4.22
GNNExplainer	0.782	0.176	22.40	21.83

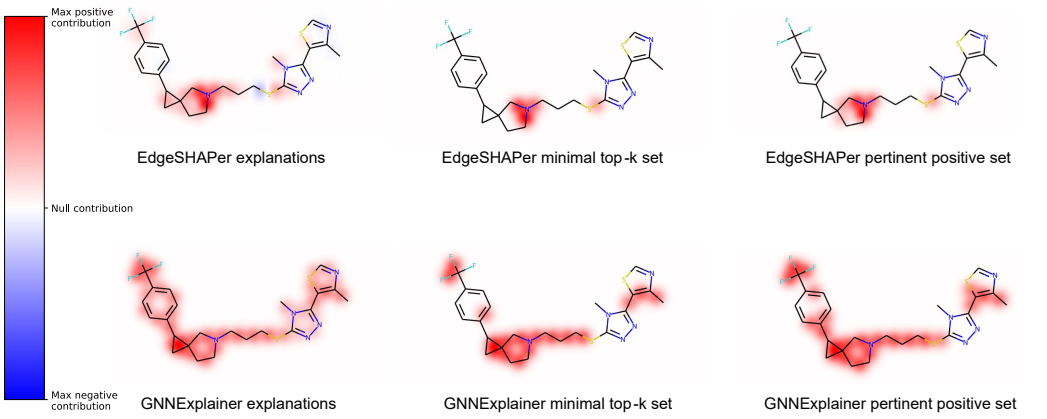
frequently covered entire compounds, making it difficult to rationalize and differentiate between predictions. As discussed above, the formation of coherent substructures by features prioritized by EdgeSHAPer that were much smaller than the ones delineated by GNNExplainer clearly indicated that chemically meaningful structural motifs were driving the predictions, as identified by EdgeSHAPer.

Taken together, the results indicated that EdgeSHAPer distinguished between bonds of different relevance for correct predictions at a higher resolution than GNNExplainer. Moreover, T_k edges found by EdgeSHAPer were critically important for the predictions. Removal of these bonds eliminated substructural coherency while determining P_{POS} edges using EdgeSHAPer revealed how salient substructures evolved, representing a high level of consistency between feature importance assessment and mapping.

We also determined the correlation between edge/bond importance derived using the different explanation methods. Since the absolute values from the different techniques could not be directly compared, we computed different rank correlation coefficients for importance-based edge rankings, including Spearman's ρ, Pearson's r, and Kendall's τ coefficients [424], as reported in Table 4.4. For both the complete ranking and the top 25% of ranked edges, correlation coefficients were generally close to 0, indicating the presence of largely distinct rankings produced with the different methods. These findings also reinforced the need for feature mapping and identification of coherent substructures determining the predictions, which are indicative of meaningful bond ensembles prioritized for model explanation, as shown for EdgeSHAPer above.



(a) Compound C#Cc1ccc2sc(C(=O)NCCCCN3CCN(c4ccccc4OC)CC3)cc2c1



(b) Compound Cc1ncsc1-c1nnc(SCCCN2CCC3(CC3c3ccc(C(F)(F)F)cc3)C2)n1C

Figure 4.4. In **(a)** and **(b)**, mappings are shown for exemplary test compounds comparing explanations from EdgeSHAPer and GNNExplainer, focusing on the minimal informative sets retrieved by the two approaches.

Table 4.4. Rank correlation coefficients for EdgeSHAPer compared to TreeExplainer and GNNExplainer for the complete ranking and the most important edges (top 25%), reported as the mean over the test set.

	Spearman's ρ	**Pearson's r**	**Kendall's τ**
Complete ranking			
TreeExplainer	0.097	0.097	0.070
GNNExplainer	−0.010	−0.010	0.022
Top 25%			
TreeExplainer	0.013	0.055	0.022
GNNExplainer	0.012	0.012	0.016

4.1.5 Computational Complexity Analysis

The computational complexity of EdgeSHAPer with Monte Carlo sampling for a single graph, without considering the complexity of the underlying neural network model, is $O(|E|^2) \cdot O(M)$, as derived below.

In Algorithm 1, the loop starting at line 2 contains operations with cost $O(N)$ (line 3) and $O(|E|)$ (lines 4, 5, 7, 8, 9, 10, 11, 12, and 13). The cost of the remaining operations is constant. Importantly, we note that the complexity of the GNN forward pass at line 14 is omitted from the analysis, given that it is highly dependent on the architecture used. Thus, operations in the loop have an asymptotic cost of $O(|N|) + O(|E|)$. Given that the loop is iterated M times, the overall cost for a single edge becomes $O(M) \cdot (O(|N|) + O(|E|))$. Furthermore, given that the number of nodes and edges in a molecular graph typically is of comparable magnitude, we can approximate $|N| \sim |E|$, obtaining an asymptotic cost of $O(M) \cdot 2O(|E|) = O(M) \cdot O(|E|)$. Since the operation must be repeated for all the edges in a molecular graph, the overall asymptotic cost is $O(|E|^2) \cdot O(M)$. For the alternative Algorithm 2 the asymptotic cost is analogous.

4.1.6 Observations

With EdgeSHAPer, we have introduced a novel methodology devised to assess the importance of edge information for GNN-based predictions. Even though GNNs are increasingly popular in many fields, including chemoinformatics and medicinal chemistry, they are among the most challenging machine learning models to explain [425]. EdgeSHAPer combines the Shapley value concept from cooperative game theory and a novel Monte Carlo sampling strategy. Shapley values determining predictions are estimated for each edge of a graph. By analyzing Shapley

value contributions, informative graph pathways can be identified. Given its edge-centric nature, EdgeSHAPer is particularly attractive for chemical applications where edges correspond to bonds connecting atoms in a molecular graph. However, EdgeSHAPer is by no means confined to rationalizing compound predictions but generally applicable to any GNN-based task.

In our proof-of-concept investigation, machine learning-based compound activity predictions were carried out and explained. Feature attributions from EdgeSHAPer were compared to a popular SHAP method for explaining decision tree models (TreeExplainer) and another edge-centric explanation method currently available, representing one of the most used XAI strategies in the field (GNNExplainer). For correct predictions, EdgeSHAPer yielded high fidelity scores and the smallest pertinent positive feature sets. Although GNNExplainer is designed to identify the subgraph determining an individual prediction, EdgeSHAPer produced smaller edge sets driving correct model decisions, leading to simpler interpretations.

Feature mapping on compound structure representations provides intuitive access to predictions for chemists. Substructures delineated by edges determining correct predictions can be interpreted in molecular terms. Such visualizations revealed the formation of coherent substructural motifs by bonds prioritized by EdgeSHAPer. The reference methods identified larger feature sets responsible for activity predictions, which often encompassed nearly complete compound structures. These findings indicated higher resolution of EdgeSHAPer explanations.

Our analysis clearly showed that GNN-based molecular predictions can be rationalized on the basis of edge/bond information rather than node/atom information, which has mostly been attempted thus far. This might be especially interesting for MPNNs centered on bonds instead of atoms, which avoid unnecessary loops during the message passing phase, as proposed for molecular property prediction [426]. Taken together, our findings indicate that EdgeSHAPer further extends the spectrum of current XAI approaches for chemical applications and beyond and should merit further consideration.

To these ends, we extended EdgeSHAPer from compound activity classification to the regression task of potency prediction for drug discovery. In this, we aimed to understand the learning characteristics of GNNs when applied to this context and unveil what they truly learn when predicting protein–ligand interaction affinities. Previous studies argued that memorization effects are in place instead of a genuine learning process of the interactions [27]. In the next section, we will delve deeper

and analyze if and to what extent this is true, discussing the applicability of deep learning for this task. The work on EdgeSHAPer was published in *iScience* by Cell Press [25].

4.2 Learning Characteristics of Graph Neural Networks Predicting Protein–Ligand Affinities

Along with determining the activity of a compound, the prediction of the potency of interaction between a ligand molecule, which is the core of a drug, and its target protein, gene, or enzyme is paramount in the research of promising hit compounds in drug design. Clearly, this presents as a more challenging task than activity prediction since the deep learning model has to predict the actual affinity value as accurately as possible. As introduced in Section 2.2.1, compound potency prediction has been tackled in machine learning with SVM and RF models [106], even if lately the panorama has been dominated by deep learning, including CNNs and RNNs [110] in both ligand-based and structure-based studies. However, more recently, the rise of graph-based models led to the employment of GNNs for potency prediction from protein–ligand interaction graphs, which are simplistic, artificially built graphs representing interactions using information from X-ray structures of protein–ligand complexes. Various GNN affinity prediction models have been reported [27, 149, 141, 142, 427], as described in Section 2.2.1. For such models, a strong correlation between predicted and experimental ligand affinities has often been observed, leading to suggestions that GNNs might be capable of learning protein–ligand interactions and associated energetics. While it is hard to conceive that physical foundations and thermodynamics of protein–ligand interactions could possibly be learned from relatively simple graph representations, the apparent prediction accuracy achieved by GNNs raised high hopes for graph-based affinity predictions in structure-based drug design. On the other hand, results of these affinity predictions are also controversially viewed [27, 428, 429]. For example, in GNN predictions, different training data volumes have often yielded similar correlation with experimental data, which is counterintuitive for deep learning, whereas different partitions of training and test data caused significant differences in model performance, also giving rise to concerns [428, 429]. Furthermore, Volkov et al. [27] have used MPNNs to predict ligand affinities from different versions of interaction graphs, including full graphs and subgraphs considering only the ligand or protein. Strikingly, MPNN models based upon only the ligand or protein graph were more accurate than models trained on full interaction graphs, indicating that the MPNNs mostly memorized ligand and protein training. Furthermore, the authors generated simple non-MPNN models that predicted the potency of ligands in complexes as the average of the

potency of most similar training set ligands. These baseline models approached the accuracy of the best graph-based models, providing corroborating evidence for memory effects in MPNN predictions [27].

In light of controversial views in the field concerning the apparent accuracy and relevance of GNN affinity predictions, we have been interested in determining what GNNs really learn when predicting protein–ligand affinity from interaction graphs. Therefore, we have systematically predicted affinities using different types of GNNs. On the basis of these results, we have applied EdgeSHAPer, now extended to regression problems, as an XAI tool to rationalize these predictions in detail, going beyond previous investigations and uncovering unexpected findings.

4.2.1 Study Concept and Methodological Framework

For our analysis, we implemented six different GNNs, systematically predicted protein–ligand affinities with each GNN, and explained the predictions to determine and compare GNN learning characteristics. GNNs with different architectures included a GCN [156], graph attention network (GAT) [430], graph isomorphism network (GIN) [431] and an edge-including variant (GINE) [432], a GNN for inductive representation learning (GraphSAGE) [302], and another GNN employing the graph convolutional (GraphConv) operator [433], termed herein GC-GNN. These networks were derived to systematically predict affinities on training (7301 curated protein–ligand complexes) and validation (658 curated complexes) sets of protein–ligand interaction graphs [27] generated from X-ray structures and affinity data available in the PDBbind database [434, 435] (details are provided later in Section 4.2.2). As test data, we employed the 2016 core set (186 curated complexes) from the PDBbind archive comprising high-quality interactions covering a wide range of affinities and the 2019 hold-out set (2542 curated complexes) [436] that were used as standards in previous investigations [118, 27]. The prediction accuracy of the different GNNs was compared to literature data reported for MPNN [27] as an immediate reference for the performance level achieved by GNNs on these datasets.

For XAI analysis, the information content of interaction graphs needs to be considered. Commonly used interaction graphs consist of edges connecting ligand pseudo-atoms and protein pseudo-atoms. Ligand pseudo-atoms represent ligand atoms or intermediate positions between atoms forming intramolecular non-covalent interactions, and protein pseudo-atoms represent amino acid residues [437, 438]. Interaction edges account for different types of non-covalent intermolecular interactions (hydrogen bonds or hydrophobic/van der Waals interactions) [437, 438]. The

assessment of edge importance is a generally applicable approach for explaining predictions based on interaction graphs. Node importance might also be considered but only implicitly accounts for interaction information. Therefore, to rationalize the prediction outcomes of different GNN models, we applied EdgeSHAPer, which is GNN model-agnostic and is able to quantify the importance of edges in graphs for GNN learning, identifying edges that are most important for individual predictions. This made it possible to directly determine which parts (subgraphs) of interaction graphs were responsible for model decisions.

The analysis with EdgeSHAPer reveals in a non-ambiguous manner to which extent GNNs learn protein–ligand interactions from graph representations for affinity predictions. GNNs can only learn information that is contained in interaction graphs. In a graph, protein–ligand interaction information is only captured by interaction edges. Therefore, if the predictions depend on learning interactions between ligands and proteins, intermolecular interaction edges must dominate the predictions. By contrast, if intramolecular ligand or protein edges make significant contributions, GNN learning focuses on ligand or protein information memorized from training data, such as structurally similar ligands having a similar affinity for the same or different proteins. Accordingly, ligand or protein memorization means that GNNs would predict affinities by recalling values associated with ligand or protein training instances without learning interactions. Figure 4.5 summarizes the workflow of the analysis. Interaction graphs were constructed from structures of protein–ligand complexes. The GNNs were derived, optimized, and evaluated using interaction graphs of the training and validation sets and then used to predict numerical affinity (pKi) values for the external core and hold-out sets. This measure represents the negative decadic logarithm of the inhibition constant ($\mathrm{pKi} = -\log_{10}(\mathrm{Ki})$). A unitary change in pKi indicates a change of an order of magnitude in Ki. High pKi values correspond to strong interactions. As a performance measure, the conventional root mean square error (RMSE) of predicted relative to experimental affinity values was calculated. Then, the EdgeSHAPer algorithm was used to quantify edge importance and identify edges (and corresponding subgraphs) determining individual predictions.

4.2.2 Data and Methods

In this section, we will illustrate the data used in the study and how they were processed. Moreover, we will describe the graph neural network architectures used.

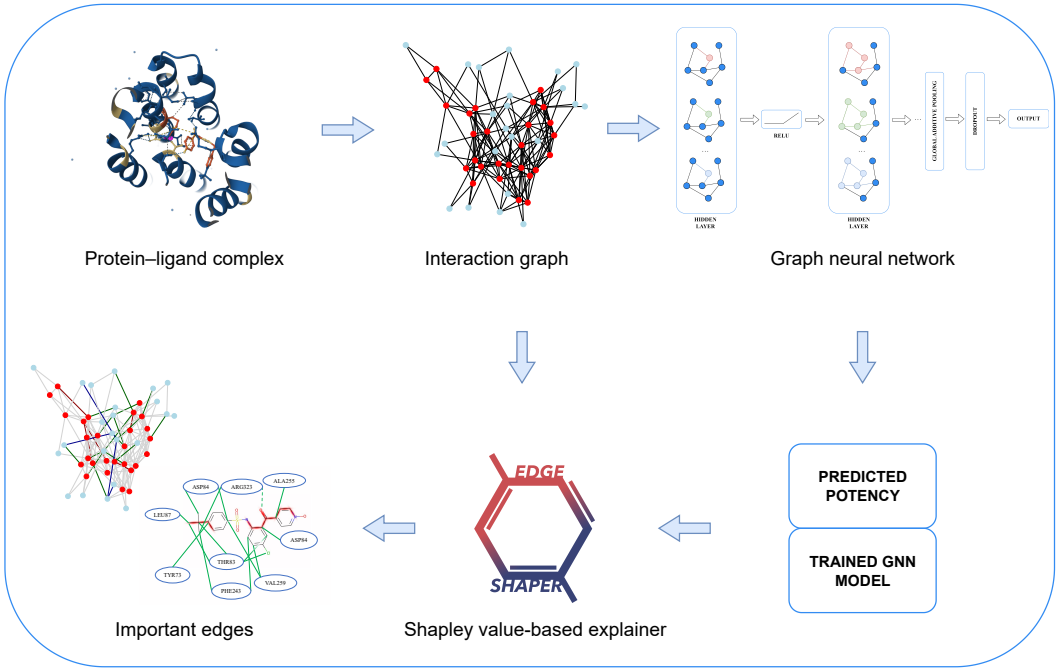


Figure 4.5. Rationalizing affinity predictions based on protein–ligand interaction graphs. The schematic representation summarizes the different stages of the analysis, including the generation of interaction graphs from X-ray structures for training and testing a GNN to predict numerical affinity values, followed by the determination of edge importance for predictions and delineation of subgraphs determining the predictions.

4.2.2.1 Structural and Affinity Data

From PDBbind (2019 version) [434, 435], training and validation sets (9962 and 903 protein–ligand complexes and associated affinities, respectively) were extracted, as described [27]. As test sets, the 2016 core and 2019 hold-out sets (257 and 3393 complexes, respectively) [436] were retrieved from PDBbind. From these non-overlapping training, validation, and test sets, complexes with error-prone affinity annotations including very low (negative logarithmic) potency values of less than 5 pKi or unusually high potency values of more than 11 pKi were removed. In addition, complexes for which the generation of interaction graphs failed [27] due to the presence of structural ambiguities were discarded. The final curated training, validation, core, and hold-out sets consisted of 7301, 658, 186, and 2542 complexes, respectively.

4.2.2.2 Protein–Ligand Interaction Graphs

For these complexes, interaction graphs made publicly available by Volkov et al. [27] were obtained, and exemplary samples were reproduced based on distances and

interaction types determined with IChem (version 5.2.9) [439] using NetworkX [440]. Following the original protocol [27], ligand pseudo-atoms and protein pseudo-atoms were defined as described below and connected via an edge if they were located within a distance of less than 6 Å. This distance threshold took into account that entire protein residues were represented as pseudo-atoms and increased the number of interaction edges compared to 4 Å, leading to lower RMSE values for predictions using interaction graphs, as reported in previous studies [27]. In addition, two ligand or protein pseudo-atoms were connected if their distance was less than 4 Å (considering both covalent and non-covalent intramolecular interactions). Edges were annotated with interaction distances that were scaled according to interquartile ranges of the global distance distribution [441]. As node features, one-hot encoded representations were used in which the one-valued entry indicated the type of non-covalent interaction involving the pseudo-atom [438, 27]: CA, hydrophobic; NZ, ionic (interacting protein atom, positively charged); N, hydrogen bond (interacting protein hydrogen-bond donor atom); OG, hydrogen bond (interacting protein hydrogen-bond acceptor and donor atom); O, hydrogen bond (interacting protein hydrogen-bond acceptor atom); CZ, aromatic; OD1, ionic (interacting protein atom, negatively charged); ZN, metal coordination.

4.2.2.3 Graph Neural Network Architectures

Different GNNs were implemented using PyTorch [421] and PyTorch Geometric (PyG) [304] libraries. The following models were generated.

Network with graph convolutional operator GC-GNN comprised seven PyG GraphConv layers representing the graph convolutional operator introduced by Morris et al. [433]. Each layer contained 256 hidden units and employed the max aggregator. The ReLU activation function was applied following each convolutional layer to introduce nonlinearity. The final network component was a global additive pooling layer, followed by a dropout layer (with a probability of 0.5) to avoid overfitting and a linear layer for regression. The GraphConv operator is defined as follows:

$$\mathbf{x}_i' = \mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2 \max_{j\in\mathcal{N}(i)} (e_{j,i}\mathbf{x}_j),$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the neural network weights, $e_{j,i}$ represents the edge weight from node j to node i, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the feature vectors for nodes i and j, respectively, and $\mathcal{N}(i)$ is the neighborhood of node i.

Graph convolutional network The GCN model we employed consisted of four convolutional layers [186], each with 256 hidden units. The ReLU activation function was used to introduce nonlinearity following each layer. After the last GCN layer, global mean pooling was carried out, followed by a dropout layer with a probability of 0.5 and a linear layer for the regression task. The definition of the GCN operator was given in Equation (4.2) when describing EdgeSHAPer (Section 4.1.3).

Graph attention network The GAT we devised was composed of seven attention layers [430], each of which contained 256 hidden channels, followed by a ReLU activation function. Global additive pooling and dropout (with a probability of 0.5) were applied. The GAT operator is defined as

$$\mathbf{x}'_i = \alpha_{i,i}\mathbf{\Theta}\mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}\mathbf{x}_j,$$

where $\alpha_{i,j}$ are the attention coefficients as defined in the original publication [430] and $\mathbf{\Theta}$ the neural network weights.

Graph isomorphism network with or without edge weights The GIN model employed four convolutional operators [431]. The GIN layer is defined as

$$\mathbf{x}'_i = h_{\mathbf{\Theta}}\left((1+\epsilon)\,\mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \mathbf{x}_j\right),$$

where ϵ is a real number as defined in the original publication [431] and $h_{\mathbf{\Theta}}$ is a neural network. In our case, a multilayer perceptron with two fully connected layers with 256 hidden channels each was used, followed by batch normalization and the ReLU activation function. Following the convolutions, a global additive pooling layer, dropout layer (with a probability of 0.5), and final linear layer for regression were added.

A network variant taking graph edge weights for the aggregation into account (GINE) [432] was also derived. In this case, the summation term becomes $\mathrm{ReLU}(\mathbf{x}_j + \mathbf{e}_{j,i})$, where $\mathbf{e}_{j,i}$ denotes the attribute (weight/distance) of the edge connecting node j to node i, transformed to match $\mathbf{x}_j$ dimensions.

GraphSAGE The GraphSAGE model is analogous as introduced in Section 3.2.1.2 with Equation (3.4) when presenting XGDAG. In this implementation, the network contained seven GraphSAGE layers with 256 hidden channels and the mean aggregation function. ReLU was used as a nonlinear activation function after each layer. The GNN architecture terminated with a global additive pooling layer, followed by a dropout layer with a probability of 0.5 and a linear layer for regression.

The different networks were trained for 100 epochs using Adam optimizer [307]. The learning rate was set to $1e-3$, weight decay to $5e-4$, and batch size to 32. During parameter optimization, the network weights were iteratively adjusted to minimize the RMSE loss function, and the models with the lowest RMSE for the validation set were selected as the final models for test set predictions. All the networks were monitored to prevent oversmoothing, as in Section 3.2.1.2.

4.2.2.4 Model Explanation

All predictions were first explained using EdgeSHAPer by identifying edges determining predictions. In each case, we used 100 sampling steps as defined in the EdgeSHAPer protocol. Edges in test graphs were ranked by Shapley values, and alternative sets of top k edges were selected by varying k from 5 to 25 (in increments of 5). Explanations were visualized using NetworkX and display features of EdgeSHAPer. Model explanations were also calculated using GNNExplainer for comparison purposes. GNNExplainer is methodologically distinct from EdgeSHAPer, as thoroughly described in this thesis. It identifies edges forming a compact subgraph that maximizes the mutual information between the network prediction and the distribution of possible subgraphs. Differently from EdgeSHAPer, GNNExplainer requires training to learn the mask and generate explanations. Hence, this approach is more hypothetical and can be sensitive to different training conditions and parameter settings. To enable an unbiased comparison, we trained GNNExplainer using the same number of epochs and learning rate as the GNN models and default parameter settings [225].

4.2.3 Predictive Performance

The GNNs were trained and evaluated on interaction graphs from the training and validation sets and then used to predict potency values for the core and hold-out sets, used as external datasets. Table 4.5 reports the performance of the GNN models compared to MPNN [27] as a reference. For GCN, GAT, GIN, GINE, GraphSAGE, and GC-GNN, lower RMSE values were obtained than reported for the reference MPNN (the training and test sets used in the independent studies were largely

overlapping but not identical to the ones used in previous work with MPNN [27]). However, the differences between these models were generally small. For the large hold-out set, the largest RMSE difference between any pair of models was only 0.397 (MPNN vs. GAT). Furthermore, RMSE differences between the different GNN architectures we investigated were very small. For the hold-out set, the largest difference between any pair of GNN models was only 0.161 (GINE vs. GAT). Thus, independently derived models achieved comparable accuracy in affinity predictions based on interaction graphs with very small differences between the models. For our XAI analysis, this was an important aspect, providing a sound basis for explaining the predictions of different GNN variants in the presence of comparable model performance.

Table 4.5. Test set predictions. For the core and hold-out sets, RMSE values are reported for GCN, GAT, GIN, GINE, GraphSAGE, GC-GNN, and reference MPNN literature data. All six GNN models reported herein were trained and evaluated on identical training, validation, and test sets.

Method	Core set RMSE	Hold-out set RMSE
MPNN	1.605	1.563
GCN	1.397	1.218
GAT	1.321	1.166
GIN	1.318	1.290
GINE	1.398	1.327
GraphSAGE	1.277	1.173
GC-GNN	1.329	1.280

4.2.4 Explanation Results

For each prediction, the k most important edges were identified using EdgeSHAPer, that is, the top k edges having the highest absolute Shapley values. Edges belonged to three different categories including (i) intramolecular edges formed between ligand pseudo-atoms (termed ligand edges), (ii) intramolecular edges formed between protein pseudo-atoms (protein edges), and (iii) intermolecular edges formed between ligand and protein pseudo-atoms (interaction edges). For model explanation, the hold-out set was used as a test set. Predictions of test instances falling into different affinity sub-ranges including low affinity (pKi < 6), medium affinity (pKi $\in [6.5, 7.5]$), and high affinity (pKi > 8) were then separately analyzed. To clearly differentiate

between these potency sub-ranges and avoid representing a continuous affinity range prone to boundary effects, test instances falling into intermittent intervals of 0.5 pKi units between these affinity sub-ranges were excluded from the analysis. The low-, medium-, and high-affinity sub-ranges that were separately analyzed contained 533, 698, and 615 test instances, respectively. The medium-affinity sub-range included the mean affinity value of all complexes (pKi = 7.15).

We determined that the proportions of ligand and protein nodes in all interaction graphs (including training and test data as well as test instances falling into different potency sub-ranges) consistently were ~60% and ~40%, respectively (with differences of only 1-2% between data sets). On average, the relative proportion of ligand and protein nodes participating in interactions was 49% and 51% across all interaction graphs, respectively. Furthermore, the relative proportion of inter- and intra-molecular edges was nearly constant across all affinity sub-ranges, with ~30% and ~70%, respectively (and only ~1% differences across the different sub-ranges). Intra-molecular edges included ~64% and ~6% ligand and protein edges, respectively. Hence, many more ligand than protein edges were available. Comparison of the proportions of ligand vs. protein pseudo-atoms and ligand vs. protein edges in interaction graphs indicated that ligand subgraphs were more densely connected than protein subgraphs, as expected (since each protein pseudo-atom can at most be connected to two others, except in the rare case of an additional disulfide bond). Using EdgeSHAPer, we then identified the edges making the largest contributions to all predictions. Figure 4.6 compares the proportions of protein, ligand, and interaction edges among the top 25 edges determining the predictions of different GNN models for the three affinity sub-ranges. These values account for the relative contributions of ligand, protein, and interaction edges to affinity predictions. All values are also reported in Table 4.6, including results from GNNExplainer for an immediate comparison.

The results in Figure 4.6 and Table 4.6 revealed clear and consistent trends for the predictions. Ligand edges dominated predictions across different affinity levels, with an average of ~65% of the top 25 edges. Protein edges made much smaller contributions, mostly only representing ~10% of the top 25 edges across all affinity sub-ranges. This might be due to the much lower propensity of protein edges than ligand edges in the graphs, as discussed above. However, all GNN models also learned interaction information, corresponding to ~20% of the top edges for four of six GNNs. Interestingly, two GNNs displayed different relative edge contributions for increasing affinity. For GIN, protein edges increased from 10.8% to 22.4% from low over medium to high affinity, and interaction edges increased from 16.8% to 30.0%,

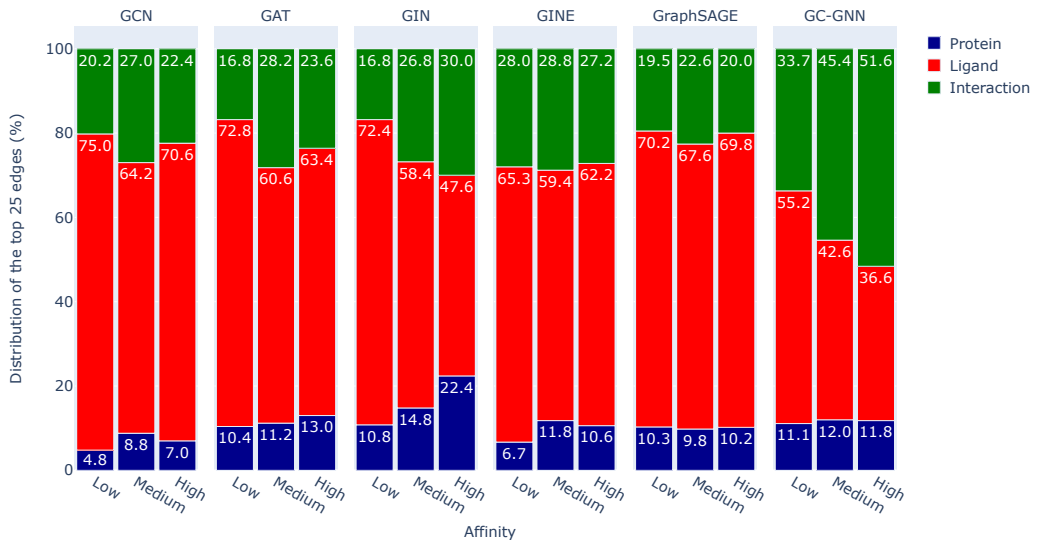


Figure 4.6. Relative proportions of edges determining predictions for different affinity sub-ranges. Color-coded bars compare mean proportions of protein, ligand, and interaction edges among the top 25 edges determining test set predictions prioritized by EdgeSHAPer. Relative proportions are separately compared for test instances falling into the three affinity sub-ranges.

whereas ligand edges decreased from 72.4% to 47.6% for increasing affinity. Moreover, for GC-GNN, protein contributions were constantly small, with only 11-12% of the top 25 edges. Ligand edges dominated low-affinity predictions, with more than half of the top edges, but then decreased when predicting increasing affinity whereas the proportion of interaction edges increased. For the high-affinity set, the top edges contained 36.6% ligand and 51.6% interaction edges. When monitored across the entire potency range, ligand and interaction edges nearly equally contributed to GC-GNN predictions with, on average, 44.8% ligand and 43.6% interaction edges, in the presence of inverse relative contributions for varying affinity.

The analysis was repeated with the conceptually distinct GNNExplainer explanation method, also capable of quantifying edge importance. For four of six networks (GAT, GINE, GraphSAGE, and GC-GNN), closely corresponding trends were detected with both explanation methods, as detailed in Table 4.6. For the remaining two networks, GCN and GIN, the explanation methods prioritized ligand and interaction edges in different ways. For GCN, GNNExplainer increasingly prioritized interaction edges from low- over medium- to high-affinity predictions (from 25.2% to 53.2%) and deprioritized ligand edges (from 69.8% to 36.0%) while EdgeSHAPer consistently prioritized ligand edges (with an average of 69.9%). For GIN, EdgeSHAPer increasingly

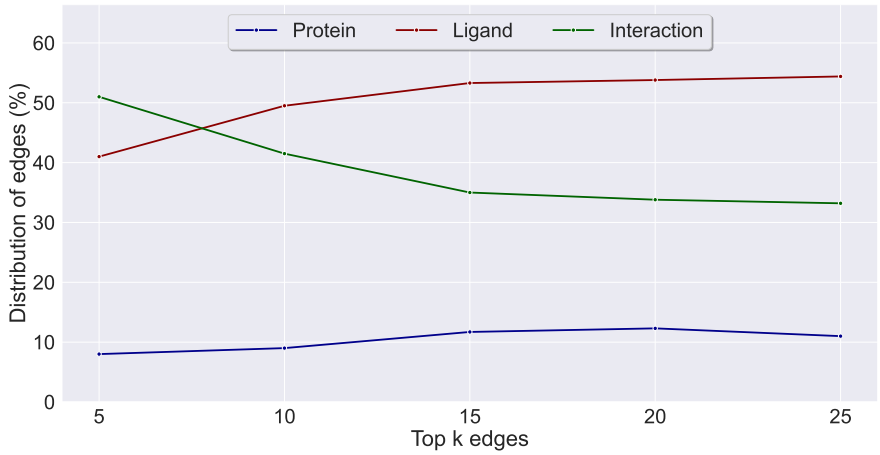
Table 4.6. Mean proportions (%) of protein, ligand, and interaction edges among the top 25 edges determining test set predictions as prioritized by EdgeSHAPer and GNNExplainer (in parentheses). The average at the bottom is calculated for all three affinity sub-ranges. In addition, the RMSE on the test set for each GNN model and sub-range is reported. All models were derived using identical training sets and tested on the same test set (hold-out set). Predictions were then separately analyzed for test instances falling into different affinity sub-ranges.

Affinity Set	Edges			RMSE
	Protein	Ligand	Interaction	
GCN				
Low	4.8% (5.0)	75.0% (69.8)	20.2% (25.2)	1.735
Medium	8.8% (7.2)	64.2% (47.0)	27.0% (45.8)	0.531
High	7.0% (10.8)	70.6% (36.0)	22.4% (53.2)	1.561
Average	6.9% (7.6)	69.9% (50.9)	23.2% (41.4)	1.276
GAT				
Low	10.4% (6.5)	72.8% (67.5)	16.8% (26.0)	1.570
Medium	11.2% (9.2)	60.6% (62.0)	28.2% (28.8)	0.541
High	13.0% (8.2)	63.4% (61.0)	23.6% (30.8)	1.563
Average	11.5% (8.0)	65.6% (63.5)	22.9% (28.5)	1.225
GIN				
Low	10.8% (7.9)	72.4% (68.0)	16.8% (24.1)	1.667
Medium	14.8% (7.0)	58.4% (67.6)	26.8% (25.4)	0.902
High	22.4% (6.2)	47.6% (71.6)	30.0% (22.2)	1.544
Average	16.0% (7.0)	59.5% (69.1)	24.5% (23.9)	1.372
GINE				
Low	6.7% (3.6)	65.3% (77.9)	28.0% (18.5)	1.583
Medium	11.8% (4.2)	59.4% (68.8)	28.8% (27.0)	0.900
High	10.6% (5.4)	62.2% (65.2)	27.2% (29.4)	1.729
Average	9.7% (4.4)	62.3% (70.6)	28.0% (25.0)	1.404
GraphSAGE				
Low	10.3% (7.1)	70.2% (70.8)	19.5% (22.1)	1.471
Medium	9.8% (7.6)	67.6% (59.4)	22.6% (33.0)	0.667
High	10.2% (8.6)	69.8% (56.4)	20.0% (35.0)	1.564
Average	10.1% (7.8)	69.2% (62.2)	20.7% (30.0)	1.234
GC-GNN				
Low	11.1% (6.9)	55.2% (52.3)	33.7% (40.8)	1.607
Medium	12.0% (6.4)	42.6% (40.4)	45.4% (53.2)	0.831
High	11.8% (6.2)	36.6% (35.6)	51.6% (58.2)	1.622
Average	11.6% (6.5)	44.8% (42.8)	43.6% (50.7)	1.353

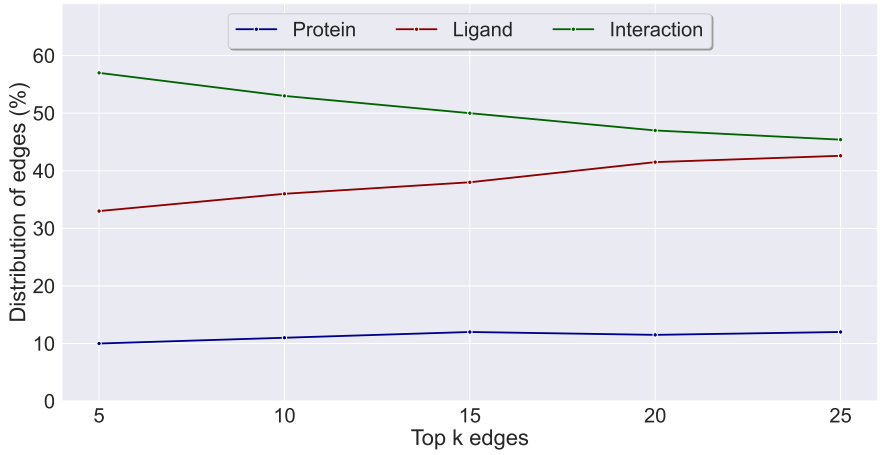
prioritized interaction edges, as discussed above, while GNNExplainer consistently prioritized ligand edges (with an average of 69.1%). Thus, EdgeSHAPer and

GNNExplainer explanations revealed affinity-dependent utilization of interaction edges for two of six networks: GIN and GC-GNN (EdgeSHAPer), GCN and GC-GNN (GNNExplainer).

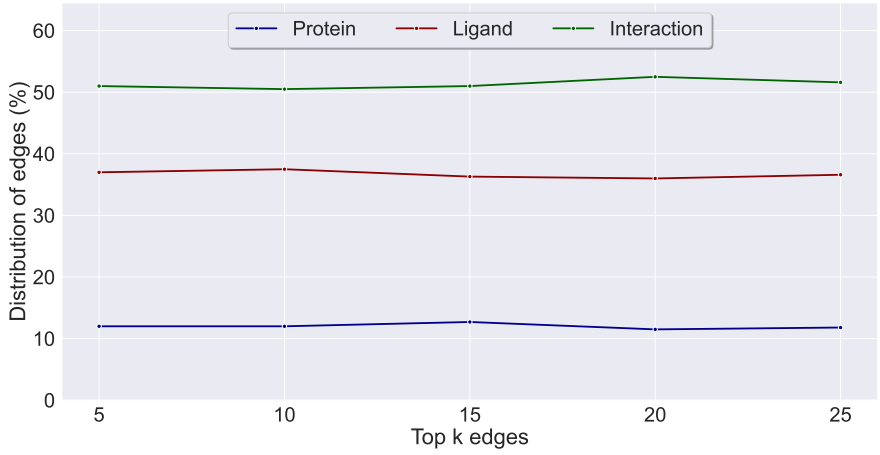
Taken together, the results of the analysis demonstrated that different types of GNNs did not consistently and exclusively learn protein–ligand interaction information from graphs to arrive at apparently accurate affinity predictions. However, interaction edges were prioritized to varying extents. In addition, while it is not known how many interaction edges might be required to correctly predict a given ligand affinity, the results showed that ligand memorization dominated the predictions overall. Accordingly, GNNs can often predict ligand affinity values with reasonable accuracy without prioritizing interaction edges and learning protein–ligand interactions by recalling affinities of similar training compounds. Notably, for increasing affinity, different relative contributions of protein, ligand, and interaction edges were observed for GNN variants with distinct architecture. Among these, the strongest systematic contributions of protein–ligand interaction edges to accurate predictions of increasing affinity were detected for GC-GNN with both explanation methods. Hence, this GNN architecture was most sensitive to the inclusion of interaction information to accurately predict high affinity. Therefore, we further analyzed the GC-GNN learning characteristics compared to other GNNs. Figure 4.7 details trends observed for GC-GNN predictions by monitoring the proportions of edges for increasing values of k, where k is the number of top edges considered.



(a) Low-affinity interactions.



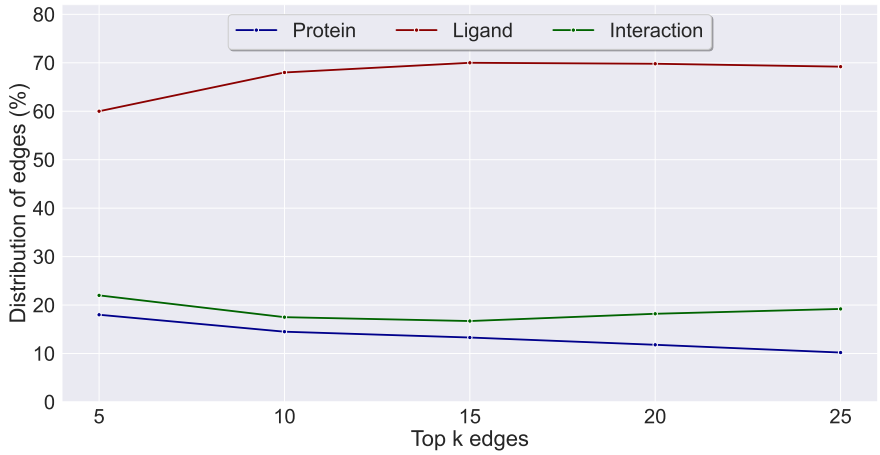
(b) Medium-affinity interactions.



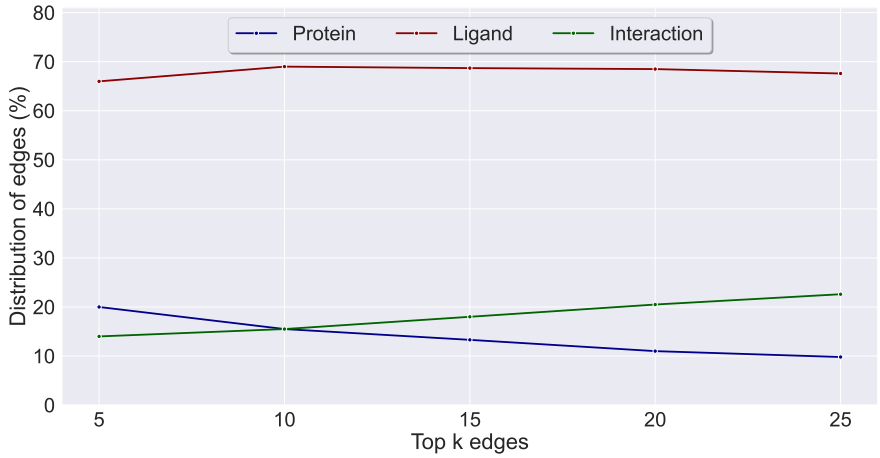
(c) High-affinity interactions.

Figure 4.7. Varying numbers of edges determining GC-GNN predictions. Monitored are the proportions of protein, ligand, and interaction edges among the top k edges for the low- **(a)**, medium- **(b)**, and high-affinity **(c)** set and increasing k values.

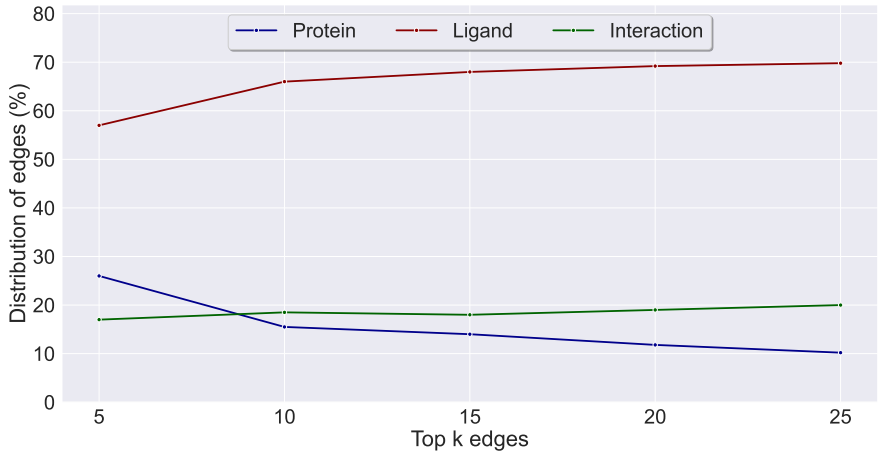
For predictions of low affinity and increasing k values (Figure 4.7a), ligand contributions began to outweigh interaction contributions when the top 10 edges were considered and then reached a plateau at the top 15 edges above 50%, dominating the predictions, while interaction contributions decreased in a mirror image-like fashion. For predictions of medium affinity and small k values, interaction contributions also dominated (Figure 4.7b). For increasing values of k, ligand contributions increased, while contributions of interactions decreased until the contributions became comparable in magnitude for the top 25 edges. By contrast, for high-affinity predictions, relative contributions of ligands and interactions to the predictions essentially remained constant over the entire range, with the largest relative contributions of interactions, followed by ligands, and only minor contributions of proteins (Figure 4.7c). For all sets, contributions of protein edges to predictions were consistently small. Consistent with the results in Figure 4.6, predictions of other networks were dominated mainly by ligand edges, as shown in Figure 4.8 for GraphSAGE as a representative example. Here, ligand edges dominated the predictions across all affinity levels, while interaction and protein edges made modest contributions. For values of k greater than 10, interaction edges became slightly more important than protein edges.



(a) Low-affinity interactions.



(b) Medium-affinity interactions.



(c) High-affinity interactions.

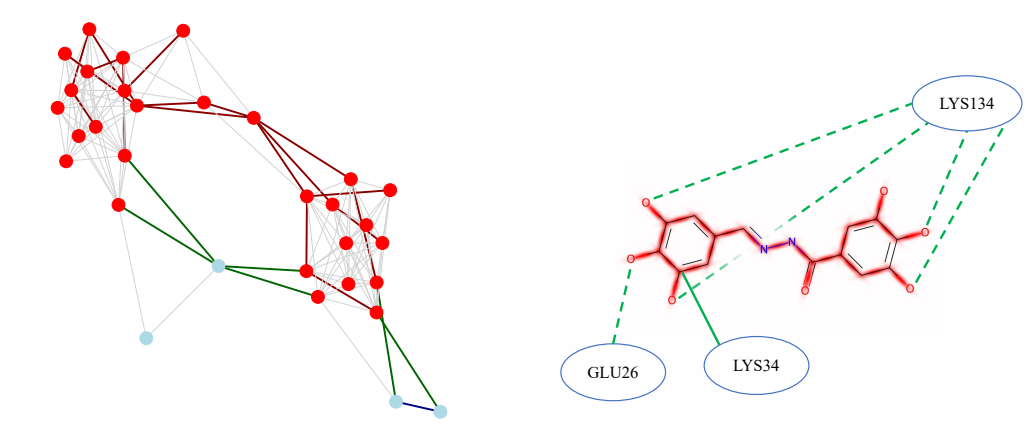
Figure 4.8. Edges determining GraphSAGE predictions. Monitored are the proportions of protein, ligand, and interaction edges among the top k edges for the low- **(a)**, medium- **(b)**, and high-affinity **(c)** set and increasing k values.

As shown in Figures 4.9 and 4.10, edges determining affinity predictions can also be mapped back from interaction graphs to actual protein–ligand interactions. Figure 4.9a shows an example explanation of a GC-GNN prediction of a low-affinity complex (pKi = 5.06) formed between a viral (H1N1) polymerase acidic endonuclease and an N-acylhydrazone inhibitor. In this case, the prediction was largely driven by ligand memorization since edges corresponding to essentially all (but one) bonds in the ligand determined the prediction. Interactions played a minor role here, mostly focusing on the side chain of LYS134 that was in hydrogen bonding distance to several ligand atoms. The dominance of ligand information was already apparent by highlighting important edges in the interaction graph and was resolved at the structural level by mapping edge information back to the protein, ligand, and protein–ligand interactions. Figure 4.9b shows the explanation of the same interaction predicted by GraphSAGE. The visualization reveals that the GraphSAGE prediction was driven by memorizing disjoint substructures of the ligand. By contrast, only two interactions involving residues GLU26 and LYS34 were detected.

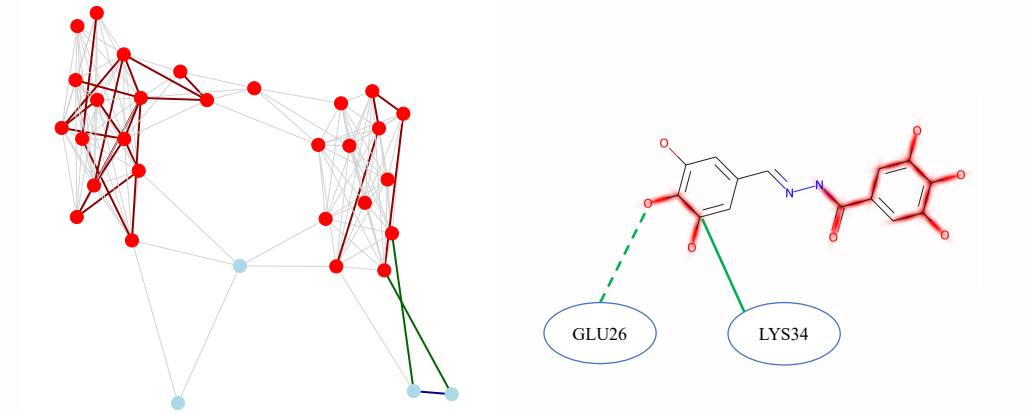
Figure 4.10 explains predictions of an exemplary high-affinity complex (pKi = 9.96) formed between the human beta chemokine receptor type 9 and a marketed antagonist (vercirnon) using GC-GNN (Figure 4.10a) and GraphSAGE (Figure 4.10b). The visualization further illustrates the different learning characteristics. For GC-GNN, the prediction was mostly determined by hydrophobic protein–ligand contacts, while ligand contributions only played a minor role. By contrast, for GraphSAGE, ligand edges again dominated the prediction, with only a minimal contribution of interaction edges different from those prioritized by GC-GNN.

4.2.5 Observations

In this section, we have investigated GNN-based affinity predictions based on protein–ligand interaction graphs from an XAI perspective. In drug design, such GNN-based affinity predictions have received increasing attention. However, the putative ability of GNNs to learn protein–ligand interactions from graphs has also been called into question, for example, by noting unusual training vs. test behavior or ligand memorization effects. Therefore, we have carried out such predictions using six GNNs with different architectures and explored underlying learning characteristics via XAI. Initially, different types of GNN models were derived and evaluated on widely used benchmark sets, reaching comparably high accuracy in the prediction of affinity values (also compared to previously reported MPNN results). By quantifying edge importance in interaction graphs, we then analyzed how these affinity predictions were determined. The interactions were classified according to different affinity



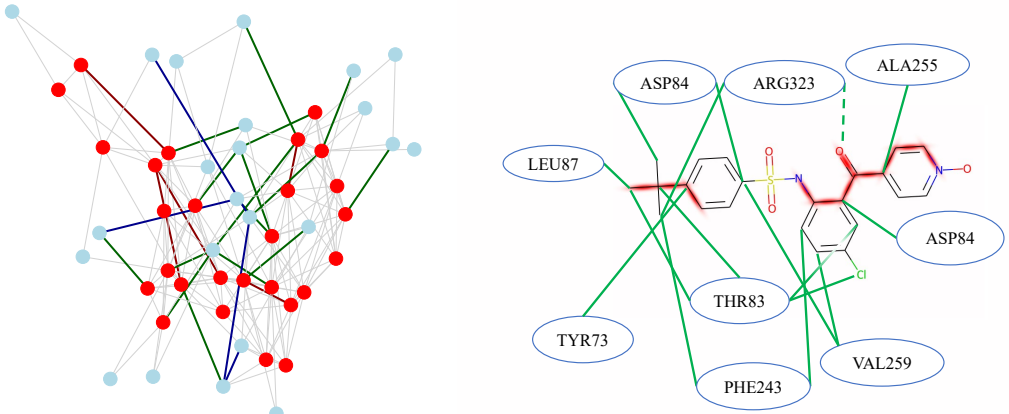
(a) Important edges - GC-GNN.



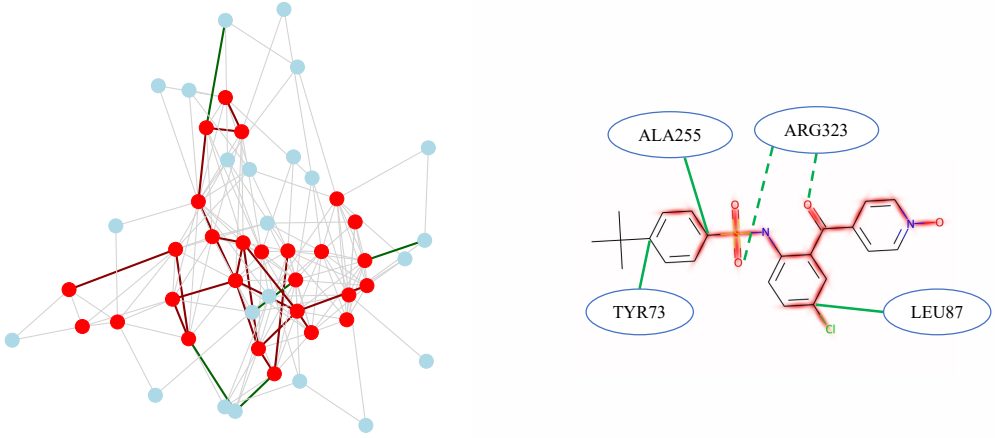
(b) Important edges - GraphSAGE.

Figure 4.9. Mapping of edges determining predictions of a low-affinity complex. For the GC-GNN prediction **(a)**, a subgraph of the interaction graph comprising the top 25 edges is displayed (left). Red and light blue nodes represent ligand and protein pseudo-atoms, respectively. Edges connecting ligand or protein pseudo-atoms are colored red and blue, respectively, and interaction edges are colored green. The top 25 edges were mapped back on ligand structure, protein structure, and protein–ligand interactions applying the same color code (right). Solid lines represent van der Waals interactions and dotted lines hydrogen bonds. For the GraphSAGE prediction of this complex **(b)**, a subgraph of the interaction graph comprising the top 25 edges is shown and the corresponding protein–ligand interactions are displayed.

levels. The results clearly showed that the GNN predictions did not exclusively depend on learning protein–ligand interactions, but that ligand memorization effects often dominated the predictions. Hence, the similarity of ligands having comparable affinity in different protein environments played a critically important role. These memorization effects can be perceived as a form of model overfitting. By contrast,



(a) Important edges - GC-GNN.



(b) Important edges - GraphSAGE.

Figure 4.10. Mapping of edges determining predictions of a high-affinity complex. A subgraph of the interaction graph comprising the top 25 edges is shown and the corresponding protein–ligand interactions are displayed for GC-GNN **(a)** and GraphSAGE **(b)**. The same color coding as Figure 4.9 applies.

protein memorization did not substantially contribute to the prediction. Importantly, however, all GNN models also learned interaction edges for affinity predictions, as revealed by edge importance analysis using two distinct XAI methods. Although the relative proportions of ligand and interaction edges varied depending on GNN model architecture, for three of six GNNs (GAT, GINE, GraphSAGE), interaction edges constituted ~20% of the most important edges for the predictions of different affinity sub-ranges. For the three other GNNs (GCN, GIN, GC-GNN), predictions of low-, medium-, and high-affinity values were determined in different ways, as revealed by explanations from EdgeSHAPer and GNNExplainer: an unexpected

finding. In these cases, model explanation identified distinct contribution patterns for predictions at different affinity levels. Low-affinity interactions were largely determined by ligand edges. For GCN and GIN, the importance of protein and interaction edges then increased for predictions of increasing affinity, whereas the importance of ligand interactions decreased. For GC-GNN, protein contributions were consistently small, but ligand and interaction edges had opposing effects on predictions of increasing affinity. For medium-affinity prediction, ligand and interaction contributions were balanced. However, for high-affinity predictions, more than half of the most important edges were interaction edges. From this point of view, GC-GNN was the most sensitive graph-based learning architecture for affinity predictions, as shown by consistent explanations from different methods.

Our findings for a variety of GNN models demonstrate that predictions of affinity based on interaction graphs do not represent showcase examples for deep learning of physical reality. However, neither are they merely "Clever Hans" predictors [442, 443], which yield promising predictions for other than anticipated reasons. Rather, the emerging picture is more differentiated. The observation that affinity predictions of different GNNs only partly depend on learning protein–ligand interaction edges from graphs is reassuring from a rigorous scientific perspective. This is the case because interaction graphs describing the endpoints of complex binding processes in a static and very simplified manner contain no thermodynamically relevant information that could potentially be exploited for learning of enthalpic or entropic effects determining ligand affinities. Instead, ligand memorization effects consistently play an important role in affinity predictions using GNNs of different architectures. The importance of ligand memorization can at least partly be attributed to the fact that structural analogues often have comparable potency (with activity cliffs being an infrequent exception). This represents an artificial feature of these predictions that strictly depends on benchmark settings that could, in principle, intrinsically limit the applicability of GNNs for prospective applications. However, our findings also demonstrate that some GNNs, such as GIN and, in particular, GC-GNN, learn interaction patterns, especially for predicting high affinities. Interaction patterns represent the most specific information contained in interaction graphs, setting interactions apart from recurrent protein and ligand information.

The observation that interaction information becomes particularly relevant for the prediction of high affinities using GNN architectures, such as GC-GNN, also offers an interesting and exciting perspective for further methodological developments. Clearly, predicting highly potent compounds is most important for practical purposes in drug design. To this end, new types of interaction graphs might be designed that

specifically emphasize interaction patterns while including only fuzzy ligand (and protein) components. For GNN predictions, these graph characteristics would work against ligand memorization and more strongly focus on interaction information determining high affinity. Thus, pairing the design of more expressive and representative interaction graphs with the ability of GNNs to learn interaction-related content can guarantee those models a deserved place in the drug development pipeline. The study described in this section was published in *Nature Machine Intelligence* [28].

4.3 A Step Toward Exact Shapley Value Computation

In Section 4.1, we have seen the development of EdgeSHAPer, our methodology that performs Shapley value approximation to determine edge importance for GNN predictions in molecular activity. Subsequently, in Section 4.2, we used it to rationalize the behavior of GNNs for interaction affinity prediction. However, the approximation of Shapley values is not a panacea for all explainability problems. For instance, there is evidence that approximation-based strategies cannot entirely capture the importance of input features when using SVMs for compound activity prediction [9]. So, in light of the work on XAI presented so far in this thesis, we thought it would be interesting to explore this aspect and try to add an exact Shapley value computation component to the proposed pipeline, enhancing its trustworthiness. Moreover, initially approaching this task by using less complicated machine learning models can offer a simplified view to tackle such a challenging problem and lay the foundations for a future extension to neural networks, as we will hint in the future perspectives in Chapter 5. Furthermore, as we introduced in Chapters 1 and 2, classic machine learning models are valid tools in chemoinformatics, and the development of XAI solutions for those is, therefore, of high scientific relevance.

As a last work presented in this thesis, we will describe SVERAD, an algorithm able to calculate Shapley values for SVMs in an exact manner relying on the radial basis function (RBF) kernel. So, we will take a step back from neural networks to take a step forward toward exact Shapley value computation.

4.3.1 Scientific Context and Motivation

As extensively remarked in this thesis, machine learning has become a key component of computer-aided drug discovery. Fast-growing volumes of chemical and biological discovery data provide a sound basis for the derivation of models for practical applications. The data deluge also causes a need for predictive modeling in support of experimental programs. In early-phase drug discovery, many machine learning applications focus on the prediction of candidate compounds with

desired biological activity. This is usually paired with the requirement to rationalize predictions for experimental design, leading to the development and usage of XAI methods to open the machine and deep learning black boxes [194, 5]. Moreover, we described the Shapley value concept [26] from collaborative game theory, adapted for quantifying feature importance in machine learning [236]. In the XAI adaptation, players correspond to features and the game is the prediction of a test instance. Given the need to enumerate and calculate the marginal contribution of a feature in each possible coalition, the computational requirements scale exponentially with increasing numbers of features. Hence, it becomes infeasible for machine learning models based on large feature sets. Therefore, approximations were introduced, as described in Section 2.3.2, and explored in this thesis with EdgeSHAPer in Section 4.1. In contrast, calculation of exact Shapley values has thus far only been accomplished for deriving local explanations of decision tree-based models [238] such as RFs, and for SVMs in combination with the Tanimoto kernel [444, 445], as recently reported [9]. The Tanimoto kernel is a similarity measure related to the Jaccard similarity and is mostly used in chemistry applications. The decision tree- and SVM-based Shapley value approaches were termed TreeSHAP (or TreeExplainer) [238], already introduced in this thesis, and Shapley Value-Expressed Tanimoto similarity (SVETA) [9], respectively. Both RF and SVM have for long been among the most popular machine learning methods in pharmaceutical research and other scientific fields, which often rival the performance of deep neural networks on sets of structured data with well-defined features (such as fingerprints) [129, 238], for example, in molecular property and activity prediction [446]. Accordingly, rationalizing SVM black-box predictions is also of considerable interest. Notably, there was only limited correlation between exact Shapley values calculated for the SVM/Tanimoto kernel combination and corresponding SHAP values, indicating that the local approximation might not be suitable for reliable model explanations in this case [9]. Given that the Tanimoto kernel is a special kernel function mostly applied to account for chemical similarity, we devised a methodology for calculating exact Shapley values for SVM models based on the more generally applied RBF kernels (including the popular Gaussian kernel). Herein, we report the development and proof-of-concept application of the Shapley-Value Expressed Radial basis function (SVERAD) approach yielding exact Shapley values for the SVM/RBF combination in a computationally efficient manner (requiring quadratic computational time with respect to the number of input features rather than exponential). Comparison of SVERAD and SHAP values revealed limited correlation, hence reinforcing the need for calculation of exact Shapley values to explain SVM predictions.

We first develop the theory and mathematical foundations of SVERAD and then

demonstrate the calculation of exact Shapley values using SVERAD based on a model system. In addition, compound activity predictions are carried out using SVM and RF models, and features determining the predictions are identified with SVERAD (SVM), KernelSHAP [8], the general applicable SHAP approximation (SVM, RF), and TreeSHAP [238] (RF). These calculations enabled a direct comparison of SVERAD and SHAP and an additional comparison of corresponding SVM and RF predictions and their explanations. Furthermore, features prioritized for SVM and RF predictions were mapped onto the structures of correctly predicted test molecular compounds to complement numerical analysis and compare chemically intuitive graphical explanations. Finally, XAI analysis is complemented by computational complexity analysis for SVERAD.

4.3.2 Data Processing and Predictive Models

For compound-based Shapley value calculations and activity predictions, we used a set of 287 adenosine receptor A3 ligands from ChEMBL [414, 415, 416] with curated high-confidence activity annotations, as reported previously [9]. As negative (inactive) examples, an equally sized set of other ChEMBL compounds was randomly selected. Compounds were represented as a keyed Morgan fingerprint [123] with bond radius 2 (that is, a binary feature vector in which each bit position represents a unique feature) [121] calculated using RDKit [419]. The fingerprint comprises compound-specific numbers of layered atom environments, which represent topological structural features [423]. Each compound is described by 5487 binary features.

Compound activity predictions were carried out using SVM and RF models, relying on the Scikit-learn implementations [422]. The data set comprising active and random compounds was divided into training (50%) and test (50%) sets. The training set was then used for grid search hyperparameter optimization via cross-validation by randomly partitioning the compounds ten times into training (50%) and validation (50%) subsets, preserving class balancing. Specifically, for the SVM models, the parameters γ (used to control nonlinearity, see later) and C were optimized. C controls the applied regularization. Smaller values of C favor generalization but increase the risk of training errors. Large values lead to a harder margin and strict misclassification penalties instead, thereby improving the classification accuracy of training samples but potentially limiting the generalization ability. After grid search optimization, the best model with $\gamma = 0.01$ and $C = 10$ produced an accuracy of 93% on the test set. As for the RF model, three hyperparameters were optimized, accounting for i) the number of decision trees, ii) the minimum number of samples required to split an internal node, and iii) the minimum number of samples required

to reach a leaf node. The parameters control overfitting, model complexity, and smoothing. The best values selected via grid search were 500, 2, and 1, respectively. The final model reached an accuracy of 92% on the test set.

The Python SHAP [8] package was used for KernelSHAP and TreeSHAP calculations. For both SVM and RF, the KernelSHAP background sample was composed of 100 randomly selected training instances. For TreeSHAP, the entire training set was used as background sample, and interventional feature perturbation was used to control input feature correlation [447].

4.3.3 Methodology

Even if introduced in Section 2.3.2 with Equation (2.1) and provided in its adapted edge-centered version in Equation (4.1), we report hereby again for immediate reference the Shapley value formula, as we will use it for the subsequent calculations:

$$\phi_f(v) = \sum_{\mathcal{S} \subseteq \mathcal{F} \setminus \{f\}} \frac{|\mathcal{S}|!\,(|\mathcal{F}| - |\mathcal{S}| - 1)!}{|\mathcal{F}|!} \left(v(\mathcal{S} \cup f) - v(\mathcal{S})\right).$$

In this formula, f is the assessed feature, $\mathcal{F}$ is the complete set of features, $\mathcal{S}$ is a coalition (a subset of $\mathcal{F} \setminus \{f\}$), and v is the coalition value.

4.3.3.1 Radial Basis Function Kernel

SVMs rely on kernel functions for implicitly mapping data distributions into higher-dimensional feature space representations if linear separation of data with different class labels is not possible in a given feature space (the so-called "kernel trick" [448]). For this purpose, alternative kernel functions can be used, depending on the particular applications. Our methodology considers the widely used RBF kernel defined as

$$K(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{d(\boldsymbol{x}, \boldsymbol{x}')^2}{2\sigma^2}},$$

where $d(\boldsymbol{x}, \boldsymbol{x}')$ is the Euclidean distance between vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$:

$$d(\boldsymbol{x}, \boldsymbol{x}') = ||\boldsymbol{x} - \boldsymbol{x}'|| = \sqrt{\sum\nolimits_i (x_i - x_i')^2}.$$

The parameter σ is a free parameter used to control the level of nonlinearity of the SVM model that will determine the decision boundary. An alternative definition of the RBF uses the parameter $\gamma = \frac{1}{2\sigma^2}$, obtaining the equivalent equation

$$K(\boldsymbol{x}, \boldsymbol{x'}) = e^{-\gamma||\boldsymbol{x}-\boldsymbol{x'}||^2}.$$

Larger values of γ will lead to a more complex decision boundary, while smaller values will render it smoother. Notably, the RBF function considered is the Gaussian RBF, as it is the most common function employed in kernelized methods and has become a standard in SVM implementations [6]. RBFs are a family of functions with radial symmetry; the Gaussian one is expressed as

$$\varphi(r) \ = e^{-\gamma r^2},$$

where r is the radial distance, which usually corresponds to the Euclidean distance (as in our case).

In pharmaceutical research, SVM models are mostly derived for molecular property predictions based on chemical structures and therefore employ structural features of compounds as input. As already seen, structural features are conventionally encoded in a binary vector format (fingerprints) [121], that is, a feature can be present or absent in test instances, corresponding to bit settings of 1 or 0, respectively. In the chemoinformatics domain, SVMs are currently essentially exclusively employed with binary fingerprint descriptors. Moreover, binary input vectors are also common for other SVM modeling tasks. Therefore, we consider the binary encoding of features as a basis for Shapley value calculations. Moreover, we define I as the number of intersecting (common) features between the two feature vectors and D as the number of features in the symmetric difference (present in either one vector or the other). N_i and N_d will be the number of intersecting and symmetric difference features in a given coalition, respectively. As we show below, the computation of Shapley values using SVERAD only relies on the number of intersecting and symmetric difference features.

4.3.3.2 Shapely Values for the Radial Basis Function Kernel

In order to express feature contributions as Shapley values via the SVERAD formalism, we first need to assess the contribution of features to the Euclidean distance. We notice that features with the same value (intersecting or absent features) do not increase the distance; in fact, $(x_i - x'_i) = 0$ if $x_i = x'_i$. Of course, this is also true for non-binary features. Then, features with different values (features with symmetric difference) increase $d(\boldsymbol{x}, \boldsymbol{x}')^2$ by $\Delta_d = (0-1)^2 = (1-0)^2 = 1$. This leads to having $d(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{N_d}$ and $d(\boldsymbol{x}, \boldsymbol{x}')^2 = N_d$, indicating that only features with symmetric difference determine the distance (and kernel) value:

$$e^{-\frac{d(\boldsymbol{x},\boldsymbol{x}')^2}{2\sigma^2}} \;=\; e^{-\frac{N_d}{2\sigma^2}} \;.$$

This allows for a fast calculation of the kernel. Now, we consider a coalition of features $\mathcal{S}$ whose value $v(\mathcal{S})$ is the RBF kernel value. If $\mathcal{S}$ contains intersecting features only ($N_d = 0$) we have $v(\mathcal{S}) = e^{-\frac{N_d}{2\sigma^2}} = e^0 = 1$. This is true for any value of I (size of the intersection). Differently, for a coalition with features with symmetric difference only (or with a mixture of intersecting and symmetric difference features), the value $v(\mathcal{S}) = e^{-\frac{N_d}{2\sigma^2}}$ must be calculated given N_d and σ (or γ), as aforementioned. Finally, for the empty coalition $\mathcal{S} = \emptyset$, we set $v(\mathcal{S}) = 0$, conforming to the Shapley value formalism for the empty set [26, 8]. As a note, the choice of setting $v(\emptyset) = 0$ is arbitrary, but will have an effect on the subsequent calculation of the Shapley values. Setting it to 0 will allow us to both simplify the calculations and conform to the Shapley value formalism. However, different values can be assigned to the empty set; this could be helpful in scenarios in which feature contribution is hard to interpret using $v(\emptyset) = 0$.

To obtain Shapley values for the RBF kernel, we need to compute the change in the kernel value when a feature from the intersection f_+, or a feature from the symmetric difference f_-, are added to the coalition $\mathcal{S}$ with N_i intersecting features and N_d features with symmetric difference. For f_+ we have

$$\Delta v_{f_+}(N_i, N_d) = e^{-\frac{N_d}{2\sigma^2}} - e^{-\frac{N_d}{2\sigma^2}} = 0. \tag{4.3}$$

Adding a feature from the intersection does not change the distance, thus not the kernel value. This is always true except if f_+ is added to the empty coalition ($v(\emptyset) = 0$). In this case, the kernel value when adding the features becomes 1 ($N_d = 0$), and so

$$\Delta v_{f_+}(0,0) = 1.$$

Then, for a symmetric difference feature f_-, we have

$$\Delta v_{f_-}(N_i, N_d) = e^{-\frac{N_d+1}{2\sigma^2}} - e^{-\frac{N_d}{2\sigma^2}}.$$

When adding a feature with symmetric difference, the squared Euclidean distance increases by 1 (as shown). The change in the kernel value must be calculated consequently. When the subtracted term represents the empty coalition, its value is set to 0.

Notably, a feature that is neither present in the intersection nor the symmetric difference is a missing feature not present in any coalition. So, its contribution to the value is 0, coherently with the missingness property of additive feature attribution methods [8] (Section 2.3.2).

Once we have computed the value change, we need to calculate the number of occurrences for each possible coalition with N_i intersecting features and N_d features with symmetric difference. For f_+ we thus consider all possible combinations of N_i elements in a set of $I - 1$ elements (the assessed feature is not part of the coalition) and N_d elements in a set of D elements:

$$C_{f_+}(N_i, N_d) = \binom{I-1}{N_i}\binom{D}{N_d}.$$

Likely, for f_- we consider all possible combinations of N_i elements in a set of I elements and N_d elements in a set of $D - 1$:

$$C_{f_-}(N_i, N_d) = \binom{I}{N_i}\binom{D-1}{N_d}.$$

Once we have all the elements, we can compute the Shapley values as the sum of the products of Δv_f, C_f, and the inverse multinomial coefficient. For an intersecting feature, the Shapley value (ϕ_f) for the RBF kernel will be computed as

$$\phi_{f_+} = \sum_{N_i=0}^{I-1}\sum_{N_d=0}^{D} \Delta v_{f_+}(N_i, N_d) \cdot C_{f_+}(N_i, N_d) \cdot \binom{I+D}{1, N_i+N_d, I+D-N_i-N_d-1}^{-1}.$$

As previously shown in Equation (4.3), $\Delta v_{f_+}(N_i, N_d)$ is always 0, except if f_+ is added to the empty coalition ($N_i = N_d = 0$). In this case, the kernel value changes from 0 to 1, thus $\Delta v_{f_+}(0,0) = 1$. So, we can easily compute the Shapley value considering only the addition to the empty coalition:

$$\begin{aligned}
\phi_{f_+} &= \Delta v_{f_+}(0,0) \cdot C_{f_+}(0,0) \cdot \binom{I+D}{1, N_i+N_d, I+D-N_i-N_d-1}^{-1} \\
&= 1 \cdot 1 \cdot \frac{(N_i+N_d)!(I+D-N_i-N_d-1)!}{(I+D)!} \\
&= \frac{(I+D-1)!}{(I+D)!}.
\end{aligned}$$

Analogously, for a symmetric difference feature, we obtain

$$\begin{aligned}
\phi_{f_-} &= \sum_{N_i=0}^{I}\sum_{N_d=0}^{D-1} \Delta v_{f_-}(N_i, N_d) \cdot C_{f_-}(N_i, N_d) \cdot \binom{I+D}{1, N_i+N_d, I+D-N_i-N_d-1}^{-1} \\
&= \sum_{N_i=0}^{I}\sum_{N_d=0}^{D-1} \left(e^{-\frac{N_d+1}{2\sigma^2}} - e^{-\frac{N_d}{2\sigma^2}}\right) \cdot \binom{I}{N_i}\binom{D-1}{N_d} \cdot \frac{(N_i+N_d)!\,(I+D-N_i-N_d-1)!}{(I+D)!} \\
&= \sum_{N_i=0}^{I}\sum_{N_d=0}^{D-1} \left(e^{-\frac{N_d+1}{2\sigma^2}} - e^{-\frac{N_d}{2\sigma^2}}\right) \cdot \frac{I!}{(I-N_i)!N_i!} \cdot \frac{(D-1)!}{(D-N_d-1)!N_d!} \\
&\quad \cdot \frac{(N_i+N_d)!\,(I+D-N_i-N_d-1)!}{(I+D)!}.
\end{aligned}$$

The computation can be further simplified by aggregating common factors. The possible coalitions to which f_- can be added include the empty coalition ($N_i = N_d = 0$, Equation (4.4)), coalitions with intersecting features only ($N_d = 0$ and $v(\mathcal{S}) = 1$, Equation (4.5)), and coalitions with intersecting and symmetric difference features, or with symmetric difference features only ($N_i \in [0, I]$ and $N_d \in [1, D-1]$, Equation (4.6)). We thus obtain

$$\phi_{f_-} = e^{-\frac{1}{2\sigma^2}} \cdot \frac{(I+D-1)!}{(I+D)!} \tag{4.4}$$

$$+ \left(e^{-\frac{1}{2\sigma^2}} - 1\right) \cdot \sum_{N_i=1}^{I} \binom{I}{N_i} \cdot \frac{N_i!\,(I+D-N_i-1)!}{(I+D)!} \tag{4.5}$$

$$+ \sum_{N_i=0}^{I} \sum_{N_d=1}^{D-1} \left(e^{-\frac{N_d+1}{2\sigma^2}} - e^{-\frac{N_d}{2\sigma^2}}\right) \cdot \binom{I}{N_i}\binom{D-1}{N_d} \tag{4.6}$$

$$\cdot \frac{(N_i+N_d)!\,(I+D-N_i-N_d-1)!}{(I+D)!}.$$

4.3.3.3 Proof of Concept

To establish initial proof of concept for the approach, we calculate Shapley values for the RBF kernel with two exemplary vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ using SVERAD:

$$\boldsymbol{x} = [1\ 0\ 0\ 1\ 0]\,,$$
$$\boldsymbol{y} = [1\ 0\ 1\ 1\ 1]\,.$$

Notably, these vectors are only used to illustrate the SVERAD calculations and do not represent (high-dimensional) molecular fingerprints. The model vectors share two features (set to 1, intersecting features), so $I = 2$, have a unique feature each (set to 0 and 1, respectively, symmetric difference), so $D = 2$, and lack a feature (set to 0). For the exemplary calculation, we set $\sigma = 1$. Table 4.7 and Table 4.8 show the steps needed to compute the Shapley values for intersecting and symmetric difference features, respectively.

As discussed, the calculation of the kernel value only depends on the number of features with symmetric difference, resulting in equation

$$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{N_d}{2}} = e^{-1} = 0.368.$$

Table 4.7. Calculation of the Shapley value for the RBF kernel for an intersecting feature.

N_i	N_d	$v(\mathcal{S})$	$v(\mathcal{S} \cup f_+)$	Δv_f	# coalitions	Inverse multinomial coefficient	Δv_f · # coalitions · inv. mult. coeff.
0	0	0	1	1	$1 \cdot 1 = 1$	$1/4 = 0.25$	0.25
0	1	$e^{-\frac{1}{2}}$	$e^{-\frac{1}{2}}$	0	$1 \cdot 2 = 2$	$1/12 = 0.083$	0
0	2	e^{-1}	e^{-1}	0	$1 \cdot 1 = 1$	$1/12 = 0.083$	0
1	0	1	1	0	$1 \cdot 1 = 1$	$1/12 = 0.083$	0
1	1	$e^{-\frac{1}{2}}$	$e^{-\frac{1}{2}}$	0	$1 \cdot 2 = 2$	$1/12 = 0.083$	0
1	2	e^{-1}	e^{-1}	0	$1 \cdot 1 = 1$	$1/4 = 0.25$	0

Table 4.8. Calculation of the Shapley value for the RBF kernel for a symmetric difference feature.

N_i	N_d	$v(\mathcal{S})$	$v(\mathcal{S} \cup f_-)$	Δv_f	# coalitions	Inverse multinomial coefficient	Δv_f · # coalitions · inv. mult. coeff.
0	0	0	$e^{-\frac{1}{2}}$	$e^{-\frac{1}{2}}$	$1 \cdot 1 = 1$	$1/4 = 0.25$	0.1516
0	1	$e^{-\frac{1}{2}}$	e^{-1}	$e^{-1} - e^{-\frac{1}{2}}$	$1 \cdot 1 = 1$	$1/12 = 0.083$	−0.0199
1	0	1	$e^{-\frac{1}{2}}$	$e^{-\frac{1}{2}} - 1$	$2 \cdot 1 = 2$	$1/12 = 0.083$	−0.0656
1	1	$e^{-\frac{1}{2}}$	e^{-1}	$e^{-1} - e^{-\frac{1}{2}}$	$2 \cdot 1 = 2$	$1/12 = 0.083$	−0.0398
2	0	1	$e^{-\frac{1}{2}}$	$e^{-\frac{1}{2}} - 1$	$1 \cdot 1 = 1$	$1/12 = 0.083$	−0.0328
2	1	$e^{-\frac{1}{2}}$	e^{-1}	$e^{-1} - e^{-\frac{1}{2}}$	$1 \cdot 1 = 1$	$1/4 = 0.25$	−0.0597

The sum of the Shapley values for all features yields the kernel value. To compute the Shapley value for a feature in the intersection and a feature with a symmetric difference, Δv_f is multiplied by the number of coalitions and the inverse multinomial coefficient, and the sum over all coalitions is calculated. Given that any feature from the same set (intersection or symmetric difference) makes the same contribution to the kernel value, we need to multiply the Shapley value obtained for one representative feature of each set by I and D to obtain the total contribution of the intersecting and symmetric difference features, respectively. In our example, the

Shapley value for an intersecting feature is 0.25, and for a feature with symmetric difference it is -0.066. The set of intersecting features ($I = 2$) yields a sum of Shapley values of 0.5, while symmetric difference features ($D = 2$) contribute to the kernel value for -0.132. Features not present in either set give no contribution. The sum of these values is 0.368, which is exactly the kernel value.

As an exemplary calculation, we consider 20 random binary vectors with a small number of features ($|F| = 15$) so that Shapley values can be computed explicitly by enumerating all possible coalitions. SVERAD yields the same Shapley values as produced by the exhaustive computation, thus demonstrating the validity of the method. This is also evident in Table 4.9, which shows a comparison of SVERAD Shapley values with the SHAP approximation (for the calculations, we set $\gamma = \frac{1}{2\sigma^2} = 1$).

Table 4.9. Comparison of exact Shapley values, SVERAD, and SHAP values using Pearson's r correlation coefficient with standard deviations.

	Exact Shapley values	SVERAD	SHAP
Exact Shapley values	1.0 ± 0.0	1.0 ± 0.0	0.72 ± 0.43
SVERAD	1.0 ± 0.0	1.0 ± 0.0	0.72 ± 0.43
SHAP	0.72 ± 0.43	0.72 ± 0.43	1.0 ± 0.0

The correlation coefficient of 1 for SVERAD Shapley values and exact Shapley values confirms that both calculations return the same values (the associated error resulting from the imprecision in the representation of very small numbers is smaller than 10^{-10}). This differs from exact Shapley values vs. SHAP, for which a Fisher-transformed correlation coefficient of 0.72 ± 0.43 is obtained, reflecting the underlying local approximation of SHAP values.

4.3.3.4 Shapley Values for Support Vector Machine Predictions

In an SVM model, the distance of a vector $\boldsymbol{x}$ from the separating hyperplane is defined by the support vectors $\boldsymbol{V}_n$ and is given by

$$dist\,(\boldsymbol{x}) = b + \sum_{n=0}^{N_v-1} y_n w_n K(\boldsymbol{x}, \boldsymbol{V}_n),$$

where N_v is the number of support vectors, y_n (-1 or 1) is the class label of the support vector $\boldsymbol{V}_n$, w_n is the weight by which the class label is scaled, and $K(\boldsymbol{x}, \boldsymbol{V}_n)$

is the kernel value comparing the support vector and the predicted instance $\boldsymbol{x}$. Finally, b is a bias value.

To compute the Shapley value for the distance for each feature f, we first substitute the kernel value with the sum of the Shapley values for the RBF kernel between vector $\boldsymbol{x}$ and the support vector $\boldsymbol{V}_n$ ($\phi_{f,n}$) and scale the sum by the label and the weight:

$$\begin{aligned} dist\left(\boldsymbol{x}\right) &= b + \sum_{n=0}^{Nv-1} y_n w_n K\left(\boldsymbol{x}, \boldsymbol{V}_n\right) = b + \sum_{n=0}^{Nv-1} y_n w_n \sum_{f=0}^{|F|-1} \phi_{f,n} \\ &= b + \sum_{f=0}^{|F|-1} \sum_{n=0}^{Nv-1} y_n w_n \phi_{f,n}. \end{aligned}$$

Then, given the additivity property of Shapley values [26] (Section 2.3.2), the Shapley value for a feature f is obtained by summing up the Shapley values of f for the RBF kernel values comparing vector $\boldsymbol{x}$ with all the support vectors, properly scaled:

$$\phi_f = \sum_{n=0}^{Nv-1} y_n w_n \phi_{f,n},$$

which gives the contribution of feature f with respect to the distance from the separating hyperplane. Finally, we consider the bias as an additional feature whose value b is its Shapley value and express the distance as

$$dist\left(\boldsymbol{x}\right) = \phi_b + \sum_{f=0}^{|F|-1} \phi_f.$$

Expressing feature contributions as log odds values The distance from the hyperplane can be transformed into probability estimates using Platt scaling [449]:

$$p\left(\boldsymbol{x}\right) = \frac{1}{1 + e^{A \cdot dist(\boldsymbol{x}) + B}}.$$

Given that Shapley values for probabilities cannot be calculated from Shapley values for the distance from the hyperplane, we need to compute the logits (log odds):

$$logit\left(p\left(\boldsymbol{x}\right)\right) = log\left(\frac{p\left(\boldsymbol{x}\right)}{1-p\left(\boldsymbol{x}\right)}\right) = \cdots = log\left(\frac{1}{e^{A\cdot dist\left(\boldsymbol{x}\right)+B}}\right) = -A\cdot dist\left(\boldsymbol{x}\right) - B.$$

We can express $dist\left(\boldsymbol{x}\right)$ as the sum of the Shapley values for the distance:

$$logit\left(p\left(\boldsymbol{x}\right)\right) = -A\cdot\left(\phi_b + \sum_{f=0}^{|F|-1}\phi_f\right) - B = -\left(A\cdot\phi_b + B\right) - \sum_{f=0}^{|F|-1} A\cdot\phi_f.$$

Logits are a linear transformation of the distance. Hence, Shapley values for the logits are obtained as a linear transformation of the Shapley values for the distance (scaling by $-A$). Moreover, by scaling ϕ_b by $-A$ and offsetting it by $-B$, the Shapley value for the additional feature is obtained, analogously to the Shapley value for the distance bias b, previously calculated. The term $-\left(A\cdot\phi_b + B\right)$ is regarded as an expected value since it does not depend on other features. The sum of the Shapley values $-\sum_{f=0}^{|F|-1} A\cdot\phi_f$ represents the difference between the actual value and the expected value, conforming to the efficiency property of the Shapley value formalism [26, 8] (Section 2.3.2). It also follows that the predictive performance of original SVM models is not affected through the Shapley value modification because it exactly accounts for the RBF kernel value, as demonstrated above, and the SVM computational classification criteria do not change.

4.3.4 Results on Compound Activity Prediction

Once we have defined how to compute exact Shapley values for the RBF kernel and then use those to derive Shapley values for SVM predictions, we present the application of the methodology in a pharmaceutically relevant context. First, we compute feature contribution to the RBF kernel for selected compounds, and then we perform a more in-depth study by calculating Shapley values for SVM predictions on active and inactive molecular compounds to determine the most crucial features for the activity or inactivity of compounds. Compared to other feature importance methods adapted for the interpretation of quantitative structure–activity relationship (QSAR) models in chemoinformatics, a hallmark of the Shapley value approach is that

it can quantify contributions of features that are present or absent in test instances to their prediction, which distinguishes it from other feature weighting approaches and renders them particularly suitable with predictions relying on molecular fingerprints. It is important to underline and make clear that an absent feature, in this context, is not a missing feature as defined by the missingness properties of additive feature attribution methods [8]. The former is a feature whose value is set to 0 in the input sample, while the latter concept applies to features that are constant, similar to the dummy feature of the original Shapley values [26], as described in Section 2.3.2. In our case, a feature is considered missing when it is missing from the input sample and all the support vectors (the feature contribution to the kernel value is 0, so, coherently, its Shapley value corresponds to 0). Indeed, this feature is neither in the intersection nor symmetric difference, so it does not take part in the calculations at all, as pointed out in Section 4.3.3.2. The feature is not part of any coalition, so it has no impact on the game, in line with the cited missingness property.

4.3.4.1 Feature Contributions to The Radial Basis Function Kernel

For a direct comparison, SVERAD and SHAP values were calculated for 50 randomly selected adenosine receptor A3 ligands that we also used for compound activity predictions. Compounds were represented using topological structural features, that is, systematically calculated pathways originating from atoms with a constant bond radius, as described in Section 4.3.2. The RBF kernel was computed for all possible compound pairs, and for each pair, exact Shapley values calculated using SVERAD were compared to corresponding SHAP values from KernelSHAP. For a value $\gamma = \frac{1}{2\sigma^2} = 0.005$ as a representative example, a mean Pearson's r correlation coefficient after Fisher transformation of 0.36 ± 0.18 was obtained. The low correlation indicated that the SHAP approximation was limited in its ability to explain RBF-based similarities and that calculation of exact Shapley values was preferred.

4.3.4.2 Rationalizing Compound Activity Predictions

To apply the SVERAD approach to pharmaceutically relevant predictions and compare model explanations for different Shapley value/SHAP calculation variants, we derived SVM and RF classification models based on distinguishing A3 ligands from other randomly selected compounds (see Section 4.3.2). The SVM and RF classifiers achieved a comparably high prediction accuracy of 93% and 92%, respectively. We then analyzed these predictions in detail.

Feature contributions to classification models For SVM predictions, Shapley values and SHAP values were calculated with SVERAD and KernelSHAP, and for RF predictions with TreeSHAP and KernelSHAP. In Table 4.10, median Pearson's r correlation coefficients are reported for feature contributions and all combinations of classification models and corresponding Shapley value/SHAP calculation methods. In addition, Figure 4.11 shows the corresponding distributions of correlation coefficients.

Table 4.10. Median Pearson's r correlation coefficient between feature contributions from different models and explanation strategies.

	SVM - SVERAD	SVM - KernelSHAP	RF - TreeSHAP	RF - KernelSHAP
SVM - SVERAD	1.000	0.120	−0.040	−0.010
SVM - KernelSHAP	0.120	1.000	0.758	0.750
RF - TreeSHAP	−0.040	0.758	1.000	0.994
RF – KernelSHAP	−0.010	0.750	0.994	1.000

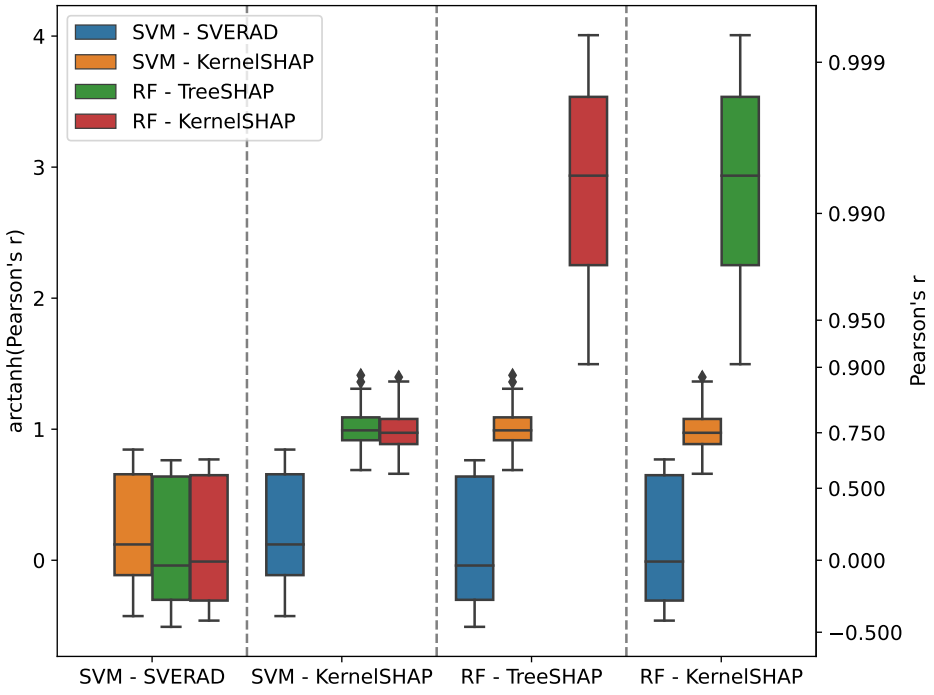
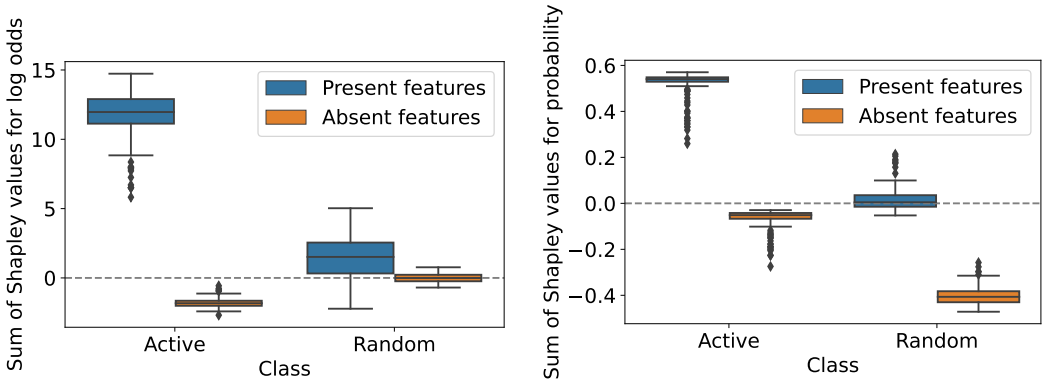


Figure 4.11. Distributions of Pearson's r correlation coefficient. Box plots represent the distributions of correlation coefficients for feature contributions from different models and explanation strategies.

There was only very low correlation between SVERAD Shapley values and SHAP

values (0.120), which reflected the limited ability of SHAP calculations to approximate Shapley values for SVM. Notably, the correlation for the SVM/RBF combination was much lower than previously determined for the SVM/Tanimoto kernel combination (0.682) [9], which clearly reinforced the need for calculating exact Shapley values if the widely applied RBF kernel is used. By contrast, for RF, there was nearly perfect correlation between KernelSHAP and TreeSHAP (0.994), which uses exact Shapley values for deriving local explanations. When comparing exact Shapley values from SVERAD and TreeSHAP for corresponding predictions, essentially no correlation was observed (−0.040), indicating that different features were determining SVM and RF predictions in the presence of comparably high prediction accuracy. However, in this case, potential correlation was also principally limited because the calculations were based on different metrics (log odds scores for SVM and class probabilities for RF). Furthermore, SHAP values for SVM and RF displayed relatively high correlation (0.758). Taken together, the results showed that SVERAD values were more accurate for SVM using the RBF kernel than the SHAP approximation, whereas TreeSHAP and KernelSHAP values were strongly correlated for RF.

Model explanations and feature mapping For the SVM and RF predictions, SVERAD and TreeSHAP values were calculated, respectively, and separately analyzed for features that were present or absent in correctly predicted test compounds. Figure 4.12 shows the distribution of cumulative Shapley values for these features in test compounds for log odds scores from SVERAD and probabilities from TreeSHAP.



(a) SVM - SVERAD. **(b)** RF - TreeSHAP.

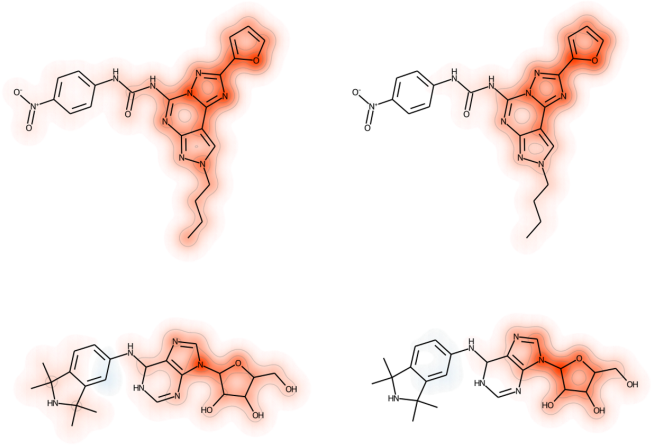
Figure 4.12. Distribution of feature contributions. Box plots show the distributions of cumulative Shapley values of features present or absent in correctly predicted test instances for SVERAD/SVM **(a)** and TreeSHAP/RF **(b)**.

The analysis explained model decisions and revealed different prediction charac-

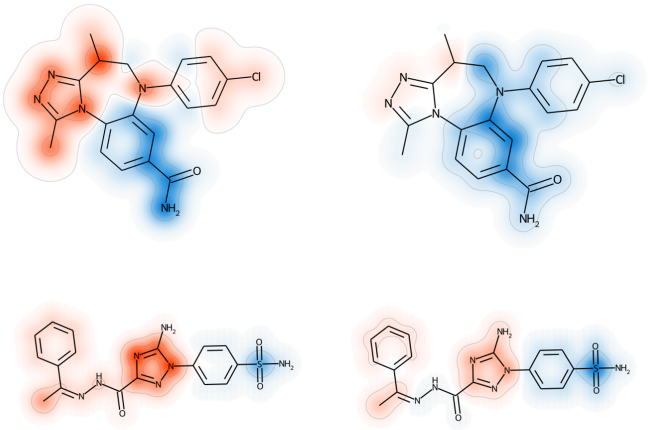
teristics for SVM and RF. For SVM, features present in active compounds made strong positive contributions to correct predictions, whereas absent features made only minor contributions to incorrect predictions. For random compounds, present features made small contributions to incorrect predictions (of activity), while absent features made essentially no contributions (with cumulative Shapley values close to zero). Hence, correct predictions of inactive compounds can only be rationalized by taking the expected value into account, as discussed in detail below. For RF, features present in active compounds determined their correct predictions, while the absence of these features in random/inactive compounds was decisive for their correct predictions. By contrast, features absent in active and present in inactive compounds made only very little or no contributions.

Overall, for active compounds, the average sum of the SVERAD Shapley values for SVM of present features was 11.65, indicating strong positive contributions to predictions far beyond the expected value (-4.61). On the contrary, absent features, with an average sum of Shapley values of -1.79, made small negative contributions. RF displayed a similar behavior for active but not for inactive compounds. Here, the average sum of the Shapley values for present and absent features was 0.51 and -0.07, respectively, and the expected value was 0.49. Accordingly, for inactive compounds, SVM predictions were largely determined by the expected value, given that features present in these compounds slightly opposed correct predictions (with average positive contributions of 1.46) while the effect of absent features was negligible (-0.008). By contrast, for RF absent features made strong contributions (-0.40 with respect to the expected value), while the average contribution of present features was only modest (0.018). Features with the largest contributions to predictions were visualized by mapping them on the corresponding atoms in correctly predicted test compounds, as shown in Figure 4.13.

For SVM and RF, features present in active compounds mostly had large positive Shapley values (red) and hence supported correct predictions (despite different value distributions, as discussed above). By contrast, for random compounds, different contributions of present features were observed. While some features supported correct predictions (blue), others opposed them (red). In active test compounds, present features supporting correct predictions with SVM and RF delineated very similar substructures.



(a) Mapping for SVM (left) and RF (right) for active compounds.



(b) Mapping for SVM (left) and RF (right) for inactive compounds.

Figure 4.13. Feature mapping. Features present in exemplary active **(a)** and random/inactive compounds **(b)** correctly predicted by SVM and RF models are mapped on corresponding atoms. For active compounds, red and blue coloring indicates positive and negative contributions toward the prediction of activity. For inactive compounds, blue indicates support toward the prediction, while red indicates opposition.

4.3.5 Computational Complexity

For SVERAD, the computation of the Shapley values for a given instance has at most $O\left(|F|^2\right)$ complexity, with $|F|$ being the total number of features.

We consider U the number of features present in the union between the explained sample and a support vector ($U = I + D$). No computation is needed for features not present in either the intersection or the difference ($\phi_f = 0$). However, in the worst case, all features are present in the union, so $U = |F|$ for all the support vectors.

For each support vector, we need to compute the Shapley value for one feature from the intersection and one feature from the symmetric difference. This computation requires $O(1)$ for the intersection (one only needs to calculate the inverse multinomial coefficient, as shown in Section 4.3.3.2), and $O(D \cdot (I + 1)) = O(D \cdot I)$ for the symmetric difference. Here, $D \cdot (I + 1)$ represents the size of the Cartesian product describing unique combinations of intersecting and symmetric difference features, also considering the empty coalition. The highest complexity would result from $D = \frac{|F|}{2}$ and $I = \frac{|F|}{2}$, leading to $O\left(\frac{|F|}{2} \cdot \frac{|F|}{2}\right) = O\left(\frac{|F|^2}{4}\right) = O\left(|F|^2\right)$.

The step above must be repeated and summed up for each support vector. Hence the complexity becomes $O\left(|F|^2\right) \cdot N_v$. Assuming the number of support vectors N_v to be a constant and given that the rest of the operations are products and sums with constant values, the final complexity will be $O\left(|F|^2\right)$.

Notably, for sparse input vectors such as for the calculations reported herein, the number of features in the union U was, on average, two orders of magnitude smaller than the total number of features $|F|$. Accordingly, the highest possible complexity is unlikely to occur in such cases. In this case, considering U as the average number of features in the union between the input sample and the support vectors, the computations require on average $O(U^2)$.

It follows that SVERAD has, at most, quadratic time requirements with respect to the number of features $|F|$ instead of the exponential computation time typically required for systematic Shapley value calculations.

As a reference measurement, executing SVERAD on the whole dataset with a machine with an Intel Core i7-12700H with 4.70 GHz of maximum clock speed and 16 GB of RAM took around 22 seconds, analogously to TreeSHAP, while running KernelSHAP calculations took more than 5.5 hours. This is further proof of the usability and computational efficiency of the proposed method.

4.3.6 Observations

As a final contribution of this thesis, we have presented SVERAD, a novel methodology for the computationally efficient calculation of exact Shapley values for SVM predictions with RBF kernels. The study follows and further extends a previous investigation determining exact Shapley values for the SVM/Tanimoto kernel combination, which is preferentially used for applications focusing on the assessment of chemical similarity. The SVM/RBF kernel combination (including the Gaussian kernel) is more widely applied. In the XAI field, the Shapley value concept experiences increasing interest in rationalizing predictions of machine learning models. Due to the complexity of explicit Shapley value calculations, approximations are typically required, for which the SHAP approach has been a pioneering development. However, low correlation between exact Shapley values calculated with SVERAD for the RBF kernel and SHAP values clearly indicated the need to use exact Shapley values for explaining SVM predictions, in marked contrast to RF. Comparative Shapley value/SHAP analysis also revealed that highly accurate SVM and RF compound predictions were determined by different relative contributions of features present or absent in active and random test compounds. However, features present in active test compounds that consistently contributed to correct predictions with both algorithms delineated corresponding substructures.

Taken together, the results reported herein indicate that SVERAD substantially aids in rationalizing SVM predictions in pharmaceutical research and other scientific fields. Therefore, in future endeavors, we aim to use SVERAD as a base to devise an exact Shapley value XAI strategy suitable also for neural networks. We will present some ideas about this extension in the next conclusive chapter. The work about SVERAD was published in *Scientific Reports* (Nature Portfolio) [29].

Chapter 5

Conclusions and Future Perspectives

Deep learning is a valuable asset that can be employed in biomedicine with compelling outcomes. However, the lack of transparency and interpretability renders its usage not broadly accepted. It is current research to understand the theory behind the superiority of neural networks, but the current results are preliminary and often limited to shallow models [450, 451]. This is why explainability techniques come into play to cope with this and provide solutions that can open the black box, explaining the results in terms of important input features.

In this thesis, we have shown the proposal of an explainable biomedical deep learning pipeline, depicted in the Introduction (Chapter 1) in Figure 1.1. In particular, we analyzed the salient components of this framework and described the proposed approaches available in the literature in Background and Related Work (Chapter 2). Then, in Chapters 3 and 4 we presented our solutions to deal with the bioinformatics and chemoinformatics elements of the pipeline, respectively.

In particular, we started our journey by describing the problem of detecting disease-associated genes (block 1 of the pipeline). Our first proposal to tackle this problem was NIAPU (Section 3.1), a network-informed adaptive positive–unlabeled learning system with a twofold outcome: delivering an accurate prioritization of disease genes while, at the same time, allowing machine and deep learning models to be effectively trained and applied in the challenging scenario of positive–unlabeled (PU) learning for disease gene discovery. Solving the issue of training in PU settings was indeed the first problem to tackle to have effective and meaningful models worth explaining. As a consequence, NIAPU plays a fundamental role in the XGDAG framework with its label propagation that enables proper learning of the graph neural network (GNN),

which otherwise would remain challenging, leading to meaningful predictions. This highlights the importance that network-based solutions hold in the proposed pipeline since they enable the practical use of deep learning models in gene prioritization.

With XGDAG (Section 3.2), we are in the second block of the pipeline, where explainability comes into play. Once the graph neural network has been properly trained thanks to the aid of NIAPU, explainable artificial intelligence (XAI) techniques are used to explain the prediction of positive (disease-associated) genes. The explanation subgraphs obtained are exploited in combination with the likely positive set by NIAPU to derive a gene prioritization ranking. The coherence of the genes detected by XGDAG was verified with an enrichment analysis, which confirmed the effective synergy of XAI and PU learning for GNNs in disease gene discovery. We showed how XAI can be used not only to explain the model's output but also as an active tool providing a ranking of candidate disease genes. XGDAG is the first method of its kind, merging XAI for graphs and PU learning in this context. As we described throughout the thesis, and especially in Section 2.1 and in Chapter 3, these techniques exploit the huge amount of omics data and gene–disease associations offered by online databases, such as BioGRID [13], STRING [47], DisGeNET [14], and eDGAR [49]. Once new associated genes are detected and verified, it is possible to integrate the discoveries back into these databases, thereby fueling future research studies. Even though already publicly available as software tools, as future development, we are planning to release a web interface to allow the use of NIAPU and XGDAG in a more straightforward and user-friendly way, increasing the reach and adoption of the systems. NIAPU was developed in collaboration with the National Research Council of Italy, and both NIAPU and XGDAG were published in *Bioinformatics* by Oxford University Press [22, 19].

In Section 3.3, we have studied that the mechanics underlying diseases may be more complex than single-gene associations and involve intricate interactions of multiple genes, which, on the contrary, may not affect the trait when considered alone, making their analysis challenging. We are referring to the genetic phenomenon of epistatic interaction. By developing our novel XAI-based pipeline, EpiDetect, that detects central genes and pathways in an epistatic network obtained via neural network explanation, we aim to offer researchers a tool to identify the most influential genetic factors and pathways that contribute to the trait of interest. This information can provide valuable insights into the underlying biology of the trait, inform the development of new treatments, and potentially lead to the discovery of novel drug targets. The newly discovered epistatic interactions can become part of gene–disease association databases, augmenting the knowledge in the field at the service of future

research. This approach has the potential to be applied to a wide range of complex traits, including cardiovascular diseases, cancer, and neurological disorders, among others. We provided evidence supporting that our proposed framework may act as a guide to disease-associated experimental research or an independent approach to validate experimental observations, increasing the possibilities offered by deep learning strategies in genetics thanks to explainability solutions. At the time of writing, this work was under consideration in a peer-reviewed journal. This research was possible thanks to a collaboration with the University of Ioannina in Greece.

Upon the identification of genes involved in a disease's etiology and regulation, it becomes possible to use those as novel drug targets. In this, both drug repurposing and de novo drug design are employed processes. We are in block 3 of our pipeline. In this thesis, we have shown how one of our methods, NIAPU, fits in a drug repurposing study (Section 3.4) focused on primary biliary cholangitis (PBC), an autoimmune liver disease orphan of treatments. NIAPU helped in augmenting the pool of genes associated with PBC, thereby providing new possible drug targets. Using new and known disease genes, we queried public databases such as DrugBank [380] to look for already-developed drugs able to target the associated genes and related pathways. The candidate drugs were relevant and meaningful for PBC, proving this approach effective and opening new possibilities for research. In fact, once new drugs are discovered or repositioned, their integration into drug–target databases offers new data for further research endeavors. Moreover, the repurposing of drugs is advantageous from both time development and safety profile perspectives. In future applications, it is possible to use XGDAG in lieu of NIAPU in the drug repositioning pipeline for a completely explainable deep learning-based drug repurposing solution. A preliminary version of this work was presented at a conference on digestive and liver diseases [452] and then, in its ultimate and complete version, published in *Biomedicines* [24] in collaboration with the National Institute of Gastroenterology "Saverio de Bellis" Research Hospital, the University of Bari, and the National Research Council of Italy.

The drug repurposing approach used for PBC is bioinformatics-driven, meaning that we relied on biological networks such as protein–protein interaction data and gene–disease associations. As we also explained in Chapter 2, drug repurposing can be perceived as a bridge task that can involve both bioinformatics and chemoinformatics tools, even employed jointly. In fact, as a prospective study, we aim to use XGDAG as a basis for a drug repurposing protocol, but we also intend to exploit knowledge graphs in the workflow. These graphs contain data gathered from different sources, ranging from gene–disease associations to drug–target links, from drug side effects to molecular structure data and genome sequences. These kinds

of graphs coalesce biological and chemical information. A GNN trained on these integrated data sources can learn novel drug–target pairs relying on knowledge that could not be exploited if working with simpler input data. By using XAI, it is possible to determine information relevant for the prediction, possibly unveiling new insights on the regulation of drug–target interactions from a holistic perspective.

After bioinformatics, in Chapter 4, we proposed chemoinformatics-based solutions to tackle the tasks of compound activity and potency prediction, which are paramount in drug design and repurposing. Given the graph-like nature of chemical data, we employed a GNN for compound activity prediction enriched by a new explainability strategy. With this, we are moving to block 4 of the pipeline, which concerns the explanation of deep learning models for drug development. In Section 4.1, we defined EdgeSHAPer, a novel edge-centric Shapley value-based explanation method for GNNs. By using a new Monte Carlo sampling strategy for graphs to approximate Shapley values, EdgeSHAPer delivered accurate and chemically relevant explanations, outperforming state-of-the-art methodologies like GNNExplainer, being the first XAI tool ever to use Shapley values for edge importance in GNNs. This work was published in *iScience* by Cell Press [25].

Notably, it is well known that accurate reporting of research methods and reproducibility of the experiments is still an open issue in scientific research [453]. Pairing this with the difficulty of using overcomplicated software tools by non-software experts may render it challenging to reproduce other scientists' results. A way to cope with this can lay in the publication of detailed research protocols that provide an in-depth description of the research methodology, from data processing to the step-by-step usage of software packages. Thus, in the spirit of open research, we published a protocol related to the use of EdgeSHAPer in *STAR Protocols* (Cell Press) [454].

In Section 4.2 EdgeSHAPer was further extended from molecular activity detection to compound potency prediction. This work was motivated by previous studies that showed controversial and thought-provoking results. On the one hand, extremely high accuracy for GNN models for affinity prediction tasks was repeatedly reported. On the other hand, there was evidence that GNNs may not properly learn protein–ligand interactions but memorize proteins and ligands with shared structures, typical of chemical datasets, to arrive at apparently high-accuracy outcomes [27]. We decided to let XAI cut the Gordian knot. After having trained and optimized several GNN models, we applied EdgeSHAPer to explain the prediction determining the salient edges defining important subgraphs. Our results were not

totally expected. If there was clear evidence that GNNs could not comprehensively learn protein–ligand interaction from simplistic graph representations of complex biochemical phenomena, we also noticed how some models prioritized interaction edges, especially for predicting high-potency compounds. This led to the conclusion that GNNs are indeed capable of detecting interaction patterns, provided they are fed with graphs containing interaction-relevant information, including only limited ligand and protein details that would work against memorization effects and favor learning behavior. For those reasons, efforts can be made in the definition of novel interaction graph representations able to leverage the power of GNNs. This research was published in *Nature Machine Intelligence* [28].

Given their ability to learn from graph data, there is still open research on GNNs and XAI. For instance, future directions can involve the development of a self-explainable GNN laying on the solid theoretical ground of Shapely values. In this, graph attention networks (GATs) can be a key. Attention coefficients in GATs can be perceived as a sort of importance metric [230]. Driving GATs to learn an approximation of the Shapley values as attention coefficients can lead to having a self-explainable model backed up by the robustness of the theory behind Shapley values.

Even though the proposed pipeline is deep-learning centered, neural networks live with other machine learning tools in the world of biomedicine. For instance, support vector machines (SVMs) are widely used tools in chemoinformatics with comparable outputs. Moreover, some of those models also present a black-box structure and need to be explained, raising the interest of researchers. Furthermore, developing XAI strategies for classic machine learning algorithms can also represent a stepping stone for future extensions to neural network models, helping in the creation of the explainable biomedical deep learning pipeline. In particular, Shapley value approximation has been widely employed to explain classic machine learning models. However, this falls short in given scenarios, like compound activity prediction using SVMs. From this, the need to obtain an exact computation of Shapley values to correctly assess input features' contribution led us to develop SVERAD, a methodology suited for SVMs with the radial basis function kernel. It delivers exact computation in quadratic time rather than exponential, rendering it a trustworthy and fast solution not only in the chemistry context but also generally applicable to any SVM prediction task. The work concerning SVERAD was published in *Scientific Reports* (Nature Portfolio) [29], and a protocol paper was underway at the time of writing, along with an extension of the methodology to additional kernels. The research related to EdgeSHAPer, SVERAD, and the protein–ligand interaction studies was carried out in a long-lasting collaboration with the University of Bonn in Germany.

As a prospective endeavor, we can use SVERAD as the starting point to devise an exact Shapley value computation strategy for deep learning models. It was demonstrated that many types of neural networks at the infinite-width limit initialized with random weights and biases correspond to Gaussian processes [455, 456, 457, 458] and that their behavior can be expressed by the neural tangent kernel [459], which is a kernel that describes the neural network evolution during the training. Those findings render it possible to use the theory behind kernel methods to analyze deep learning models and understand them in a more in-depth and theoretical manner. Ideally, even if challenging, by using insights from SVERAD, it would be interesting to devise a Shapley value computation strategy that leverages the kernel of the neural network Gaussian process in a similar manner as seen in SVERAD in order to compute Shapley values for features in neural networks.

In this thesis, we have seen all the components of the proposed explainable biomedical deep learning pipeline, with the exception of block 5: generative drug design. Instead of relying on known drugs or molecular compounds serving as the basis for de novo drug design, generative development goes beyond. By using neural network models, new compounds and drugs are designed in silico, thereby facilitating and speeding up pharmaceutical research in an unprecedented way. It is current research to use deep generative approaches to devise new compounds and drugs [30]. Many strategies like recurrent neural networks, variational autoencoders, and generative adversarial networks demonstrated their effectiveness in generative drug design [460, 461]. However, to properly find their place in our pipeline, those models must be explainable. Efforts are starting to be made for the purpose, but only a few examples are found in literature, and research in this area is far from being mature [462, 463]. For this reason, there is high scientific interest in developing XAI strategies for generative models, especially in biomedicine. Moreover, explainability strategies like EdgeSHAPer and SVERAD can drive the development of deep generative models based on expert knowledge. The important features or molecular structures identified in block 4 of the pipeline can be used to constrain generative models to obey some specific rules and include molecular components that are known to be relevant for the activity or the presence of a desired property, leading to a more accurate and effective generative drug design, backed up by previous studies. Finally, the novel-generated molecules can find their place in chemical databases, sustaining further research and filling the last gap in the explainable biomedical deep learning pipeline.

To conclude, this thesis was a journey through the world of biomedicine from the point

of view of explainable artificial intelligence for deep learning models. We proposed a pipeline that, going from bioinformatics to chemoinformatics, demonstrated how every component was fundamental for its coherent functioning. Network-based and classic machine learning solutions pave the way for more complex and advanced deep learning strategies. Deep learning alone is powerful, but its predictions cannot be trusted, given the black-box character of neural networks. This is why explainability is the cornerstone of this thesis. We presented different XAI solutions working at every step of the pipeline, thereby enhancing the trustworthiness of neural networks in bioinformatics and chemoinformatics, giving birth to new research ideas and going toward explainable biomedical deep learning.

Appendix A

Marginal Effect Analysis for DBP and PP

We hereby report additional results for the marginal effect analysis of the EpiCID component of EpiDetect (Section 3.3) for diastolic blood pressure (DBP) and pulse pressure (PP), which yield analogous observations as the analysis on systolic blood pressure (SBP) (Section 3.3.3.3). Figure A.1 and Table A.1 show the distribution of the top 5 SNPs with the highest degree in the top-1000 network for DBP.

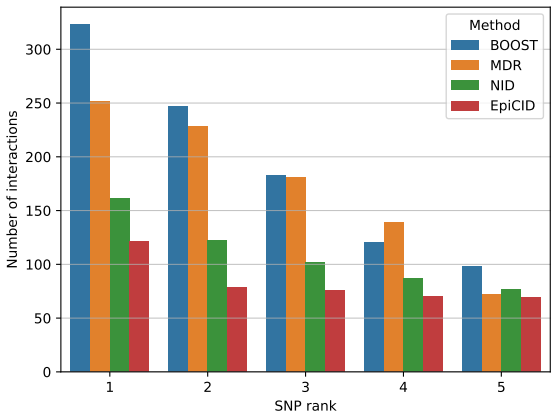


Figure A.1. Graphical representation of the distribution of the 5 highest-degree SNPs in the first 1000 interactions for DBP.

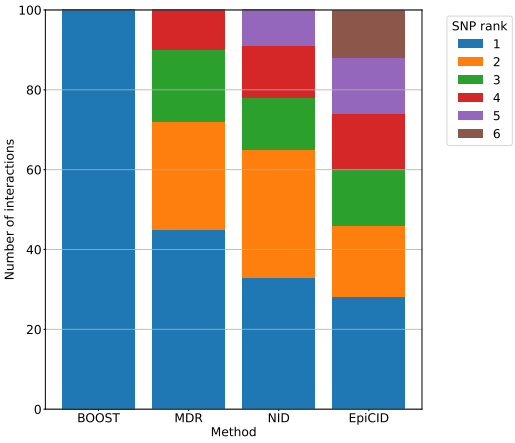
Analogously to the analysis reported in Section 3.3.3.3, we notice how neural network-based approaches, especially EpiCID, show a more variable ranking than the compared strategies and are less affected by marginal effects. Comparing the different algorithms with respect to the distribution of the SNPs with the highest

Table A.1. Distributions of the 5 most interacting SNPs in first 1000 interactions for DBP (the number of interactions corresponds to the degree in the top-1000 interaction network) and the total number of interactions in which they are involved.

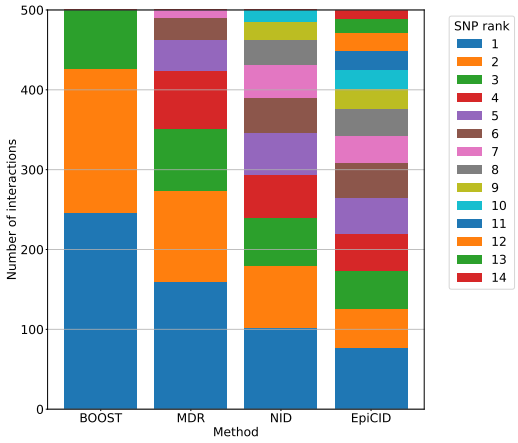
BOOST			
Rank	**SNP**	**Gene**	**Interactions**
1	rs1821295	AC011518.1	323
2	rs7134060	CDK17	247
3	rs4364717	MTAP	183
4	rs12579720	LINC02398	120
5	rs12374077	SENP2	98
Total number of interactions			971
MDR			
Rank	**SNP**	**Gene**	**Interactions**
1	rs1378942	CSK	252
2	rs3184504	SH2B3	228
3	rs16998073	FGF5	181
4	rs167479	RGL3	139
5	rs6429422	SDCCAG8	72
Total number of interactions			872
NID			
Rank	**SNP**	**Gene**	**Interactions**
1	rs12184466	CCDC63	161
2	rs13107325	SLC39A8	122
3	rs3861113	DACH1	102
4	rs12563539	RPS27	87
5	rs11026586	AC055878.1	77
Total number of interactions			472
EpiCID			
Rank	**SNP**	**Gene**	**Interactions**
1	rs73033340	ZFAND2A	121
2	rs6681713	MIR4421	79
3	rs75902664	SLC7A2	76
4	rs751984	LRRC10B	70
5	rs3861113	DACH1	69
Total number of interactions			415

number of interactions at different levels (top 100, 500, and 1000 interactions), shows that the top interactions feature the presence of few high-degree SNPs for BOOST and MDR, while more SNPs are present in NID and EpiCID rankings, confirming the analysis reported for SBP. This can be seen in Figure A.2, which shows stacked bar charts visualizing the most interacting SNPs in the top-100, top-500, and top-1000 interaction networks.

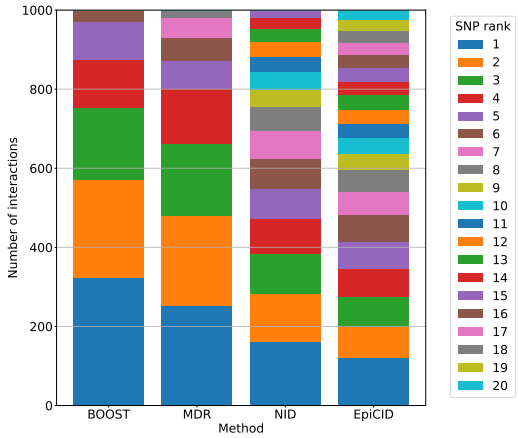
Analogous results were obtained for PP, reported in Table A.2. The analysis confirms that EpiCID is less affected by the presence of marginal effects of single SNPs, given



(a) Top 100 interactions.



(b) Top 500 interactions.



(c) Top 1000 interactions.

Figure A.2. Highest-degree SNPs in the top 100 **(a)**, top 500 **(b)**, and top 1000 **(c)** interactions for DBP.

that the top 5 SNPs are involved in a smaller number of interactions than the rest of the methods. This can also be observed in Figure A.3, which plots the distribution of the top 5 interacting SNPs in the top 1000 interactions.

Finally, Figure A.4 complements the analysis by showing the distributions of the SNPs with the highest number of interactions in the top-100, top-500, and top-1000 interaction networks, which confirm the observations reported so far.

Table A.2. Distributions of the 5 most interacting SNPs in first 1000 interactions (the number of interactions corresponds to the degree in the top-1000 interaction network) for PP and the total number of interactions in which they are involved. *Note that the total number of interactions may be more than 1000 since an interaction between two of the top 5 SNPs is counted twice (once for each SNP).

BOOST			
Rank	**SNP**	**Gene**	**Interactions**
1	rs832890	AC069368.1	212
2	rs10732433	NEBL	160
3	rs4553000	UBAP1	149
4	rs2978456	SLC20A2	139
5	rs2400509	SPINK7	92
Total number of interactions			752
MDR			
Rank	**SNP**	**Gene**	**Interactions**
1	rs4811601	KIAA1755	282
2	rs452036	MYH6	282
3	rs17287293	AC087312.1	282
4	rs12705090	TRIP6	151
5	rs11154027	RNU4-35P	10
Total number of interactions			1007*
NID			
Rank	**SNP**	**Gene**	**Interactions**
1	rs114275780	AL355499.1	178
2	rs17287293	AC087312.1	98
3	rs7977311	FGD6	63
4	rs138877676	SPIB	63
5	rs10418305	NOTCH3	56
Total number of interactions			458
EpiCID			
Rank	**SNP**	**Gene**	**Interactions**
1	rs114275780	AL355499.1	130
2	rs62270945	GATA2	103
3	rs17287293	AC087312.1	85
4	rs2498323	HGFAC	60
5	rs12705090	TRIP6	59
Total number of interactions			437

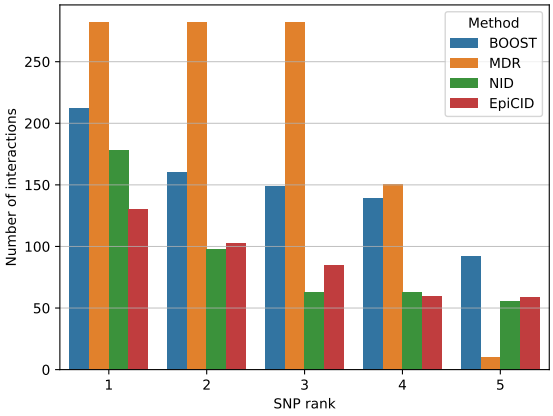
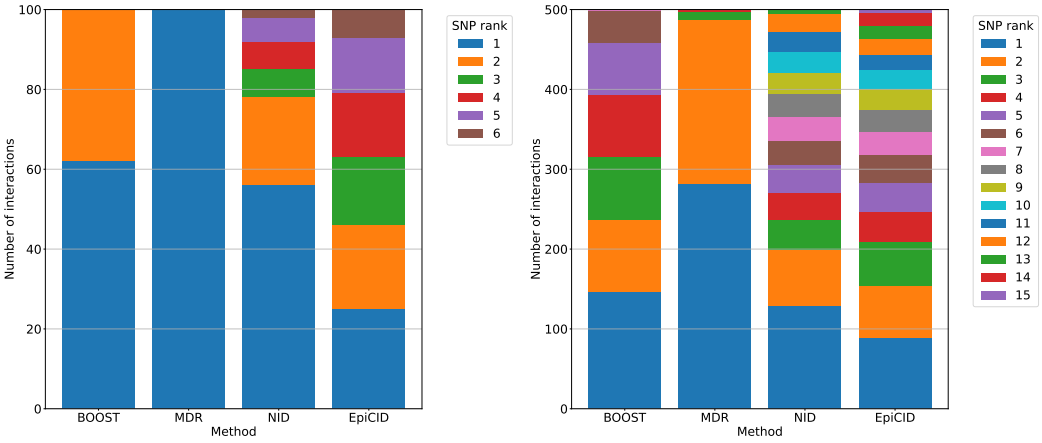
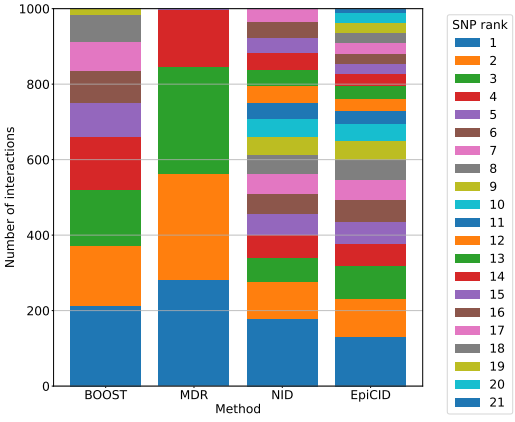


Figure A.3. Graphical representation of the distribution of the 5 highest-degree SNPs in the first 1000 interactions for PP.



(a) Top 100 interactions.

(b) Top 500 interactions.



(c) Top 1000 interactions.

Figure A.4. Highest-degree SNPs in the top 100 **(a)**, top 500 **(b)**, and top 1000 **(c)** interactions for PP.

Bibliography

[1] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, **521** (2015), 436–444.

[2] Wang, F., Kaushal, R., and Khullar, D. Should health care demand interpretable artificial intelligence or accept "black box" medicine? *Annals of Internal Medicine*, **172** (2020), 59–60.

[3] Quinn, T. P., Jacobs, S., Senadeera, M., Le, V., and Coghlan, S. The three ghosts of medical ai: can the black-box present deliver? *Artificial Intelligence in Medicine*, **124** (2022), 102158.

[4] Novakovsky, G., Dexter, N., Libbrecht, M. W., Wasserman, W. W., and Mostafavi, S. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, **24** (2023), 125–137.

[5] Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., and Yang, G.-Z. XAI — explainable artificial intelligence. *Science Robotics*, **4** (2019), eaay7120.

[6] Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, **20** (1995), 273–297.

[7] Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. Support vector regression machines. *Advances in neural information processing systems*, **9** (1996).

[8] Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, **30** (2017).

[9] Feldmann, C. and Bajorath, J. Calculation of exact Shapley values for support vector machines with Tanimoto kernel enables model interpretation. *iScience*, **25** (2022), 105023.

[10] Moreau, Y. and Tranchevent, L.-C. Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, **13** (2012), 523–536.

[11] Zolotareva, O. and Kleine, M. A survey of gene prioritization tools for mendelian and complex human diseases. *Journal of Integrative Bioinformatics*, **16** (2019), 20180069.

[12] Barabási, A.-L., Gulbahce, N., and Loscalzo, J. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, **12** (2011), 56–68.

[13] Oughtred, R., et al. The BioGRID interaction database: 2019 update. *Nucleic Acids Research*, **47** (2019), D529–D541.

[14] Piñero, J., Ramírez-Anguita, J. M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., and Furlong, L. I. The DisGeNET knowledge platform for disease genomics: 2019 update. *Nucleic Acids Research*, **48** (2020), D845–D855.

[15] Le, D.-H. Machine learning-based approaches for disease gene prediction. *Briefings in Functional Genomics*, **19** (2020), 350–363.

[16] Uffelmann, E., Huang, Q. Q., Munung, N. S., De Vries, J., Okada, Y., Martin, A. R., Martin, H. C., Lappalainen, T., and Posthuma, D. Genome-wide association studies. *Nature Reviews Methods Primers*, **1** (2021), 59.

[17] Wang, M. H., Cordell, H. J., and Van Steen, K. Statistical methods for genome-wide association studies. In *Seminars in Cancer Biology*, vol. 55, p. 53–60. Elsevier (2019).

[18] Bycroft, C., et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature*, **562** (2018), 203–209.

[19] Mastropietro, A., De Carlo, G., and Anagnostopoulos, A. XGDAG: explainable gene–disease associations via graph neural networks. *Bioinformatics*, **39** (2023), btad482.

[20] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, **20** (2008), 61–80.

[21] Bekker, J. and Davis, J. Learning from positive and unlabeled data: a survey. *Machine Learning*, **109** (2020), 719–760.

[22] Stolfi, P., Mastropietro, A., Pasculli, G., Tieri, P., and Vergni, D. NIAPU: network-informed adaptive positive-unlabeled learning for disease gene identification. *Bioinformatics*, **39** (2023), btac848.

[23] VanderWeele, T. J. Epistatic interactions. *Statistical Applications in Genetics and Molecular Biology*, **9** (2010).

[24] Shahini, E., Pasculli, G., Mastropietro, A., Stolfi, P., Tieri, P., Vergni, D., Cozzolongo, R., Pesce, F., and Giannelli, G. Network proximity-based drug repurposing strategy for early and late stages of primary biliary cholangitis. *Biomedicines*, **10** (2022), 1694.

[25] Mastropietro, A., Pasculli, G., Feldmann, C., Rodríguez-Pérez, R., and Bajorath, J. EdgeSHAPer: bond-centric Shapley value-based explanation method for graph neural networks. *iScience*, **25** (2022), 105043.

[26] Shapley, L. S. A value for n-person games. In *Contributions to the Theory of Games II* (edited by H. W. Kuhn and A. W. Tucker), p. 307–317. Princeton University Press (1953).

[27] Volkov, M., Turk, J.-A., Drizard, N., Martin, N., Hoffmann, B., Gaston-Mathé, Y., and Rognan, D. On the frustration to predict binding affinities from protein–ligand structures with deep neural networks. *Journal of Medicinal Chemistry*, **65** (2022), 7946–7958.

[28] Mastropietro, A., Pasculli, G., and Bajorath, J. Learning characteristics of graph neural networks predicting protein–ligand affinities. *Nature Machine Intelligence*, **5** (2023), 1427–1436.

[29] Mastropietro, A., Feldmann, C., and Bajorath, J. Calculation of exact Shapley values for explaining support vector machine models using the radial basis function kernel. *Scientific Reports*, **13** (2023), 19561.

[30] Zeng, X., et al. Deep generative molecular design reshapes drug discovery. *Cell Reports Medicine*, **3** (2022), 100794.

[31] Vert, J.-P. How will generative ai disrupt data science in drug discovery? *Nature Biotechnology*, **41** (2023), 750.

[32] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems*, **27** (2014).

[33] Abbasi, M., et al. Designing optimized drug candidates with generative adversarial network. *Journal of Cheminformatics*, **14** (2022), 40.

[34] Ferruz, N., Schmidt, S., and Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, **13** (2022), 4348.

[35] Chen, H. and Bajorath, J. Designing highly potent compounds using a chemical language model. *Scientific Reports*, **13** (2023), 7412.

[36] Perez-Riverol, Y., et al. Discovering and linking public omics data sets using the Omics Discovery Index. *Nature Biotechnology*, **35** (2017), 406–409.

[37] Perez-Riverol, Y., et al. Quantifying the impact of public omics data. *Nature Communications*, **10** (2019), 3512.

[38] Heather, J. M. and Chain, B. The sequence of sequencers: the history of sequencing DNA. *Genomics*, **107** (2016), 1–8.

[39] Wang, Z., Gerstein, M., and Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, **10** (2009), 57–63.

[40] Krassowski, M., Das, V., Sahu, S. K., and Misra, B. B. State of the field in multi-omics research: from computational needs to data mining and sharing. *Frontiers in Genetics*, **11** (2020), 610798.

[41] Halu, A., De Domenico, M., Arenas, A., and Sharma, A. The multiplex network of human diseases. *NPJ Systems Biology and Applications*, **5** (2019), 15.

[42] Visscher, P. M., Wray, N. R., Zhang, Q., Sklar, P., McCarthy, M. I., Brown, M. A., and Yang, J. 10 years of GWAS discovery: biology, function, and translation. *The American Journal of Human Genetics*, **101** (2017), 5–22.

[43] Pulst, S. M. Genetic linkage analysis. *Archives of Neurology*, **56** (1999), 667–672.

[44] Umlai, U.-K. I., Bangarusamy, D. K., Estivill, X., and Jithesh, P. V. Genome sequencing data analysis for rare disease gene discovery. *Briefings in Bioinformatics*, **23** (2022), bbab363.

[45] Lee, L. Y.-H. and Loscalzo, J. Network medicine in pathobiology. *The American Journal of Pathology*, **189** (2019), 1311–1326.

[46] Luck, K., et al. A reference map of the human binary protein interactome. *Nature*, **580** (2020), 402–408.

[47] Szklarczyk, D., et al. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, **49** (2021), D605–D612.

[48] Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., García-García, J., Sanz, F., and Furlong, L. I. DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, **45** (2016), D833–D839.

[49] Babbi, G., Martelli, P. L., Profiti, G., Bovo, S., Savojardo, C., and Casadio, R. eDGAR: a database of disease-gene associations with annotated relationships among genes. *BMC Genomics*, **18** (2017), 25–34.

[50] Ghiassian, S. D., Menche, J., and Barabási, A.-L. A disease module detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLoS Computational Biology*, **11** (2015), e1004120.

[51] Petti, M., Bizzarri, D., Verrienti, A., Falcone, R., and Farina, L. Connectivity significance for disease gene prioritization in an expanding universe. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **17** (2019), 2155–2161.

[52] Quinodoz, M., Royer-Bertrand, B., Cisarova, K., Di Gioia, S. A., Superti-Furga, A., and Rivolta, C. DOMINO: using machine learning to predict genes associated with dominant disorders. *The American Journal of Human Genetics*, **101** (2017), 623–629.

[53] Enright, A. J., Van Dongen, S., and Ouzounis, C. A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, **30** (2002), 1575–1584.

[54] Sun, P. G., Gao, L., and Han, S. Prediction of human disease-related gene clusters by clustering analysis. *International Journal of Biological Sciences*, **7** (2011), 61.

[55] Köhler, S., Bauer, S., Horn, D., and Robinson, P. N. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, **82** (2008), 949–958.

[56] Valdeolivas, A., Tichit, L., Navarro, C., Perrin, S., Odelin, G., Levy, N., Cau, P., Remy, E., and Baudot, A. Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics*, **35** (2019), 497–505.

[57] Petsko, G. A. Guilt by association. *Genome Biology*, **10** (2009), 104.

[58] Guney, E. and Oliva, B. Exploiting protein-protein interaction networks for genome-wide disease-gene prioritization. *PLOS ONE*, **7** (2012), e43557.

[59] Chen, J., Xu, H., Aronow, B. J., and Jegga, A. G. Improved human disease candidate gene prioritization using mouse phenotype. *BMC Bioinformatics*, **8** (2007), 392.

[60] Chen, J., Aronow, B. J., and Jegga, A. G. Disease candidate gene identification and prioritization using protein interaction networks. *BMC Bioinformatics*, **10** (2009), 73.

[61] Chen, J., Bardes, E. E., Aronow, B. J., and Jegga, A. G. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Research*, **37** (2009), W305–W311.

[62] Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank citation ranking: bring order to the web. Tech. rep., Technical Report, Stanford University (1998).

[63] Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, **46** (1999), 604–632.

[64] Mordelet, F. and Vert, J.-P. ProDiGe: prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics*, **12** (2011), 389.

[65] Mordelet, F. and Vert, J.-P. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters*, **37** (2014), 201–209.

[66] Scott, C. and Blanchard, G. Novelty detection: unlabeled data definitely help. In *Artificial Intelligence and Statistics*, vol. 5, p. 464–471. PMLR (2009).

[67] Yang, P., Li, X.-L., Mei, J.-P., Kwoh, C.-K., and Ng, S.-K. Positive-unlabeled learning for disease gene identification. *Bioinformatics*, **28** (2012), 2640–2647.

[68] Yang, P., Li, X., Chua, H.-N., Kwoh, C.-K., and Ng, S.-K. Ensemble positive unlabeled learning for disease gene identification. *PLOS ONE*, **9** (2014), e97079.

[69] Miko, I. Epistasis: gene interaction and phenotype effects. *Nature Education*, **1** (2008), 197.

[70] Niel, C., Sinoquet, C., Dina, C., and Rocheleau, G. A survey about methods dedicated to epistasis detection. *Frontiers in Genetics*, **6** (2015), 285.

[71] Hahn, L. W., Ritchie, M. D., and Moore, J. H. Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. *Bioinformatics*, **19** (2003), 376–382.

[72] Wan, X., Yang, C., Yang, Q., Xue, H., Fan, X., Tang, N. L., and Yu, W. BOOST: a fast approach to detecting gene-gene interactions in genome-wide case-control studies. *The American Journal of Human Genetics*, **87** (2010), 325–340.

[73] Jiang, R., Tang, W., Wu, X., and Fu, W. A random forest approach to the detection of epistatic interactions in case-control studies. *BMC Bioinformatics*, **10** (2009), S65.

[74] Yoshida, M. and Koike, A. SNPInterForest: a new method for detecting epistatic interactions. *BMC Bioinformatics*, **12** (2011), 469.

[75] Beam, A. L., Motsinger-Reif, A., and Doyle, J. Bayesian neural networks for detecting epistasis in genetic association studies. *BMC Bioinformatics*, **15** (2014), 368.

[76] Brown, F. K. et al. Chemoinformatics: what is it and how does it impact drug discovery. *Annual Reports in Medicinal Chemistry*, **33** (1998), 375–384.

[77] Bunin, B. A., Siesel, B., Morales, G., and Bajorath, J. *Chemoinformatics theory*. Springer (2007).

[78] Polanski, J. Chemoinformatics. In *Comprehensive Chemometrics* (edited by S. D. Brown, R. Tauler, and B. Walczak), pp. 459–506. Elsevier (2009).

[79] Schneider, G. Computational medicinal chemistry. *Future Medicinal Chemistry*, **3** (2011), 393–394.

[80] Weininger, D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, **28** (1988), 31–36.

[81] Heller, S. R., McNaught, A., Pletnev, I., Stein, S., and Tchekhovskoi, D. InChI, the IUPAC international chemical identifier. *Journal of Cheminformatics*, **7** (2015), 23.

[82] Gad, S. QSAR. In *Encyclopedia of Toxicology (Third Edition)* (edited by P. Wexler), pp. 1–9. Academic Press (2014).

[83] Meng, X.-Y., Zhang, H.-X., Mezei, M., and Cui, M. Molecular docking: a powerful approach for structure-based drug discovery. *Current Computer-Aided Drug Design*, **7** (2011), 146–157.

[84] Durrant, J. D. and McCammon, J. A. Molecular dynamics simulations and drug discovery. *BMC Biology*, **9** (2011), 71.

[85] Vamathevan, J., et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, **18** (2019), 463–477.

[86] Korotcov, A., Tkachenko, V., Russo, D. P., and Ekins, S. Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets. *Molecular Pharmaceutics*, **14** (2017), 4462–4475.

[87] Askr, H., Elgeldawi, E., Aboul Ella, H., Elshaier, Y. A., Gomaa, M. M., and Hassanien, A. E. Deep learning in drug discovery: an integrative review and future challenges. *Artificial Intelligence Review*, **56** (2023), 5975–6037.

[88] Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., and Blaschke, T. The rise of deep learning in drug discovery. *Drug Discovery Today*, **23** (2018), 1241–1250.

[89] Hughes, J. P., Rees, S., Kalindjian, S. B., and Philpott, K. L. Principles of early drug discovery. *British Journal of Pharmacology*, **162** (2011), 1239–1249.

[90] Mohs, R. C. and Greig, N. H. Drug discovery and development: role of basic biological research. *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, **3** (2017), 651–657.

[91] Ma, J., Wang, J., Ghoraie, L. S., Men, X., Liu, L., and Dai, P. Network-based method for drug target discovery at the isoform level. *Scientific Reports*, **9** (2019), 13868.

[92] Cao, C. and Moult, J. GWAS and drug targets. *BMC Genomics*, **15** (2014), S5.

[93] Avery, V. M., Camp, D., Carroll, A. R., Jenkins, I. D., and Quinn, R. J. The identification of bioactive natural products by high throughput screening (HTS). In *Comprehensive Natural Products III (Third Edition)*, p. 410–429. Elsevier (2010).

[94] Macarron, R., et al. Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery*, **10** (2011), 188–195.

[95] Attene-Ramos, M., Austin, C., and Xia, M. High throughput screening. In *Encyclopedia of Toxicology (Third Edition)* (edited by P. Wexler), pp. 916–917. Academic Press (2014).

[96] Szymański, P., Markowicz, M., and Mikiciuk-Olasik, E. Adaptation of high-throughput screening in drug discovery—toxicological screening tests. *International Journal of Molecular Sciences*, **13** (2011), 427–452.

[97] Lee, M.-Y., Park, C. B., Dordick, J. S., and Clark, D. S. Metabolizing enzyme toxicology assay chip (MetaChip) for high-throughput microscale toxicity analyses. *Proceedings of the National Academy of Sciences*, **102** (2005), 983–987.

[98] Lee, M.-Y., Kumar, R. A., Sukumaran, S. M., Hogg, M. G., Clark, D. S., and Dordick, J. S. Three-dimensional cellular microarray for high-throughput toxicology assays. *Proceedings of the National Academy of Sciences*, **105** (2008), 59–63.

[99] Lavecchia, A. and Di Giovanni, C. Virtual screening strategies in drug discovery: a critical review. *Current Medicinal Chemistry*, **20** (2013), 2839–2860.

[100] Kimber, T. B., Chen, Y., and Volkamer, A. Deep learning in virtual screening: recent applications and developments. *International Journal of Molecular Sciences*, **22** (2021), 4435.

[101] M Honorio, K., L Moda, T., and D Andricopulo, A. Pharmacokinetic properties and in silico ADME modeling in drug discovery. *Medicinal Chemistry*, **9** (2013), 163–176.

[102] Khan, A., et al. Combined drug repurposing and virtual screening strategies with molecular dynamics simulation identified potent inhibitors for SARS-CoV-2 main protease (3CLpro). *Journal of Biomolecular Structure and Dynamics*, **39** (2021), 4659–4670.

[103] ElHefnawi, M., Jo, E., Tolba, M. M., Fares, M., Yang, J., Shahbaaz, M., and Windisch, M. P. Drug repurposing through virtual screening and

in vitro validation identifies tigecycline as a novel putative HCV polymerase inhibitor. *Virology*, **570** (2022), 9–17.

[104] Gan, J.-h., Liu, J.-x., Liu, Y., Chen, S.-w., Dai, W.-t., Xiao, Z.-X., and Cao, Y. DrugRep: an automatic virtual screening server for drug repurposing. *Acta Pharmacologica Sinica*, **44** (2023), 888–896.

[105] Akamatsu, M. Current state and perspectives of 3D-QSAR. *Current Topics in Medicinal Chemistry*, **2** (2002), 1381–1394.

[106] Lewis, R. A. and Wood, D. Modern 2D QSAR for drug discovery. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, **4** (2014), 505–522.

[107] Breiman, L. Random forests. *Machine learning*, **45** (2001), 5–32.

[108] Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., and Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, **43** (2003), 1947–1958.

[109] Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications. *Drug Discovery Today*, **20** (2015), 318–331.

[110] Kim, J., Park, S., Min, D., and Kim, W. Comprehensive survey of recent drug discovery using deep learning. *International Journal of Molecular Sciences*, **22** (2021), 9983.

[111] Bajorath, J. Deep machine learning for computer-aided drug design. *Frontiers in Drug Discovery*, **2** (2022), 829043.

[112] Guedes, I. A., Pereira, F. S., and Dardenne, L. E. Empirical scoring functions for structure-based virtual screening: applications, critical aspects, and challenges. *Frontiers in Pharmacology*, **9** (2018), 1089.

[113] Liu, J. and Wang, R. Classification of current scoring functions. *Journal of Chemical Information and Modeling*, **55** (2015), 475–482.

[114] Li, H., Sze, K.-H., Lu, G., and Ballester, P. J. Machine-learning scoring functions for structure-based virtual screening. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, **11** (2021), e1478.

[115] Gleeson, M. P. and Gleeson, D. QM/MM calculations in drug discovery: a useful method for studying binding phenomena? *Journal of Chemical Information and Modeling*, **49** (2009), 670–677.

[116] Williams-Noonan, B. J., Yuriev, E., and Chalmers, D. K. Free energy methods in drug design: prospects of "alchemical perturbation" in medicinal chemistry. *Journal of Medicinal Chemistry*, **61** (2018), 638–649.

[117] Jiménez, J., Skalic, M., Martinez-Rosell, G., and De Fabritiis, G. KDEEP: protein–ligand absolute binding affinity prediction via 3D-convolutional neural networks. *Journal of Chemical Information and Modeling*, **58** (2018), 287–296.

[118] Stepniewska-Dziubinska, M. M., Zielenkiewicz, P., and Siedlecki, P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, **34** (2018), 3666–3674.

[119] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, p. 1263–1272. PMLR (2017).

[120] Deng, Z., Chuaqui, C., and Singh, J. Structural interaction fingerprint (SIFt): a novel method for analyzing three-dimensional protein- ligand binding interactions. *Journal of Medicinal Chemistry*, **47** (2004), 337–344.

[121] Maggiora, G., Vogt, M., Stumpfe, D., and Bajorath, J. Molecular similarity in medicinal chemistry. *Journal of Medicinal Chemistry*, **57** (2014), 3186–3204.

[122] Zagidullin, B., Wang, Z., Guan, Y., Pitkänen, E., and Tang, J. Comparative analysis of molecular fingerprints in prediction of drug combination effects. *Briefings in Bioinformatics*, **22** (2021), bbab291.

[123] Morgan, H. L. The generation of a unique machine description for chemical structures - a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, **5** (1965), 107–113.

[124] Zhao, C., Zhang, H., Zhang, X., Liu, M., Hu, Z., and Fan, B. Application of support vector machine (SVM) for prediction toxic activity of different data sets. *Toxicology*, **217** (2006), 105–119.

[125] Sorgenfrei, F. A., Fulle, S., and Merget, B. Kinome-wide profiling prediction of small molecules. *ChemMedChem*, **13** (2018), 495–499.

[126] Sato, T., Honma, T., and Yokoyama, S. Combining machine learning and pharmacophore-based interaction fingerprint for in silico screening. *Journal of Chemical Information and Modeling*, **50** (2010), 170–185.

[127] Ballester, P. J. and Mitchell, J. B. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, **26** (2010), 1169–1175.

[128] Rodríguez-Pérez, R., Vogt, M., and Bajorath, J. Support vector machine classification and regression prioritize different structural features for binary compound activity and potency value prediction. *ACS Omega*, **2** (2017), 6371–6379.

[129] Rodríguez-Pérez, R. and Bajorath, J. Explainable machine learning for property predictions in compound optimization. *Journal of Medicinal Chemistry*, **64** (2021), 17744–17752.

[130] Siemers, F. M. and Bajorath, J. Differences in learning characteristics between support vector machine and random forest models for compound classification revealed by Shapley value analysis. *Scientific Reports*, **13** (2023), 5983.

[131] Haykin, S. *Neural networks: a comprehensive foundation*. Prentice Hall PTR (1994).

[132] Tian, K., Shao, M., Wang, Y., Guan, J., and Zhou, S. Boosting compound-protein interaction prediction by deep learning. *Methods*, **110** (2016), 64–72.

[133] Li, F., et al. Deep neural network classifier for virtual screening inhibitors of (S)-adenosyl-l-methionine (SAM)-dependent methyltransferase family. *Frontiers in Chemistry*, **7** (2019), 324.

[134] Wallach, I., Dzamba, M., and Heifets, A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, **abs/1510.02855** (2015). `arXiv:1510.02855`.

[135] Skalic, M., Martínez-Rosell, G., Jiménez, J., and De Fabritiis, G. PlayMolecule BindScope: large scale CNN-based virtual screening on the web. *Bioinformatics*, **35** (2019), 1237–1238.

[136] Öztürk, H., Özgür, A., and Ozkirimli, E. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, **34** (2018), i821–i829.

[137] Öztürk, H., Olmez, E. O., and Özgür, A. WideDTA: prediction of drug-target binding affinity. *CoRR*, **abs/1902.04166** (2019). `arXiv:1902.04166`.

[138] Li, Y., Rezaei, M. A., Li, C., and Li, X. DeepAtom: a framework for protein-ligand binding affinity prediction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 303–310. IEEE (2019).

[139] Lee, I., Keum, J., and Nam, H. DeepConv-DTI: prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Computational Biology*, **15** (2019), e1007129.

[140] Karimi, M., Wu, D., Wang, Z., and Shen, Y. DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, **35** (2019), 3329–3338.

[141] Xiong, J., Xiong, Z., Chen, K., Jiang, H., and Zheng, M. Graph neural networks for automated de novo drug design. *Drug Discovery Today*, **26** (2021), 1382–1393.

[142] Son, J. and Kim, D. Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. *PLOS ONE*, **16** (2021), e0249404.

[143] Lim, J., Ryu, S., Park, K., Choe, Y. J., Ham, J., and Kim, W. Y. Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation. *Journal of Chemical Information and Modeling*, **59** (2019), 3981–3988.

[144] Feinberg, E. N., et al. PotentialNet for molecular property prediction. *ACS Central Science*, **4** (2018), 1520–1530.

[145] Gao, K. Y., Fokoue, A., Luo, H., Iyengar, A., Dey, S., Zhang, P., et al. Interpretable drug target prediction using deep neural representation. In *IJCAI*, vol. 2018, p. 3371–3377 (2018).

[146] Torng, W. and Altman, R. B. Graph convolutional neural networks for predicting drug-target interactions. *Journal of Chemical Information and Modeling*, **59** (2019), 4131–4149.

[147] Jiang, M., Li, Z., Zhang, S., Wang, S., Wang, X., Yuan, Q., and Wei, Z. Drug–target affinity prediction using graph neural network and contact maps. *RSC Advances*, **10** (2020), 20701–20712.

[148] Bianchi, F. M., Grattarola, D., Livi, L., and Alippi, C. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44** (2021), 3496–3507.

[149] Shen, H., Zhang, Y., Zheng, C., Wang, B., and Chen, P. A cascade graph convolutional network for predicting protein–ligand binding affinity. *International Journal of Molecular Sciences*, **22** (2021), 4023.

[150] Ashenden, S. K. Lead optimization. In *The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry*, p. 103–117. Elsevier (2021).

[151] Pushpakom, S., et al. Drug repurposing: progress, challenges and recommendations. *Nature Reviews Drug Discovery*, **18** (2019), 41–58.

[152] Jarada, T. N., Rokne, J. G., and Alhajj, R. A review of computational drug repositioning: strategies, approaches, opportunities, challenges, and directions. *Journal of Cheminformatics*, **12** (2020), 46.

[153] Lamb, J., et al. The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, **313** (2006), 1929–1935.

[154] Vidović, D., Koleti, A., and Schürer, S. C. Large-scale integration of small molecule-induced genome-wide transcriptional responses, kinome-wide binding affinities and cell-growth inhibition profiles reveal global trends characterizing systems-level drug action. *Frontiers in Genetics*, **5** (2014), 342.

[155] Subramanian, A., et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, **171** (2017), 1437–1452.

[156] Keiser, M. J., et al. Predicting new molecular targets for known drugs. *Nature*, **462** (2009), 175–181.

[157] David, L., Thakkar, A., Mercado, R., and Engkvist, O. Molecular representations in AI-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, **12** (2020), 56.

[158] Dudley, J. T., Deshpande, T., and Butte, A. J. Exploiting drug–disease relationships for computational drug repositioning. *Briefings in Bioinformatics*, **12** (2011), 303–311.

[159] Campillos, M., Kuhn, M., Gavin, A.-C., Jensen, L. J., and Bork, P. Drug target identification using side-effect similarity. *Science*, **321** (2008), 263–266.

[160] Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., and Kanehisa, M. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, **24** (2008), i232–i240.

[161] Kinnings, S. L., Liu, N., Buchmeier, N., Tonge, P. J., Xie, L., and Bourne, P. E. Drug discovery using chemical systems biology: repositioning the safe medicine comtan to treat multi-drug and extensively drug resistant tuberculosis. *PLoS Computational Biology*, **5** (2009), e1000423.

[162] Wu, C., Gudivada, R. C., Aronow, B. J., and Jegga, A. G. Computational drug repositioning through heterogeneous network clustering. *BMC Systems Biology*, **7** (2013), S6.

[163] Tan, F., et al. Drug repositioning by applying 'expression profiles' generated by integrating chemical structure similarity and gene semantic similarity. *Molecular BioSystems*, **10** (2014), 1126–1138.

[164] Martinez, V., Navarro, C., Cano, C., Fajardo, W., and Blanco, A. DrugNet: network-based drug–disease prioritization by integrating heterogeneous data. *Artificial Intelligence in Medicine*, **63** (2015), 41–49.

[165] Rakshit, H., Chatterjee, P., and Roy, D. A bidirectional drug repositioning approach for parkinson's disease through network-based inference. *Biochemical and Biophysical Research Communications*, **457** (2015), 280–287.

[166] Yang, C. C. and Zhao, M. Mining heterogeneous network for drug repositioning using phenotypic information extracted from social media and pharmaceutical databases. *Artificial Intelligence in Medicine*, **96** (2019), 80–92.

[167] Hogan, A., et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, **54** (2021), 71.

[168] Ioannidis, V. N., Song, X., Manchanda, S., Li, M., Pan, X., Zheng, D., Ning, X., Zeng, X., and Karypis, G. DRKG - drug repurposing knowledge graph for Covid-19. `https://github.com/gnn4dr/DRKG/` (2020).

[169] Al-Saleem, J., Granet, R., Ramakrishnan, S., Ciancetta, N. A., Saveson, C., Gessner, C., and Zhou, Q. Knowledge graph-based approaches to drug repurposing for COVID-19. *Journal of Chemical Information and Modeling*, **61** (2021), 4058–4067.

[170] Zhang, R., Hristovski, D., Schutte, D., Kastrin, A., Fiszman, M., and Kilicoglu, H. Drug repurposing for COVID-19 via knowledge graph completion. *Journal of Biomedical Informatics*, **115** (2021), 103696.

[171] Bang, D., Lim, S., Lee, S., and Kim, S. Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers. *Nature Communications*, **14** (2023), 3570.

[172] Gottlieb, A., Stein, G. Y., Ruppin, E., and Sharan, R. PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology*, **7** (2011), 496.

[173] Menden, M. P., Iorio, F., Garnett, M., McDermott, U., Benes, C. H., Ballester, P. J., and Saez-Rodriguez, J. Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLOS ONE*, **8** (2013), e61318.

[174] Napolitano, F., Zhao, Y., Moreira, V. M., Tagliaferri, R., Kere, J., D'Amato, M., and Greco, D. Drug repositioning: a machine-learning approach through data integration. *Journal of Cheminformatics*, **5** (2013), 30.

[175] Yang, J., Li, Z., Fan, X., and Cheng, Y. Drug–disease association and drug-repositioning predictions in complex diseases using causal inference–probabilistic matrix factorization. *Journal of Chemical Information and Modeling*, **54** (2014), 2562–2569.

[176] Lim, H., Poleksic, A., Yao, Y., Tong, H., He, D., Zhuang, L., Meng, P., and Xie, L. Large-scale off-target identification using fast and accurate dual regularized one-class collaborative filtering and its application to drug repurposing. *PLoS Computational Biology*, **12** (2016), e1005135.

[177] Ozsoy, M. G., Özyer, T., Polat, F., and Alhajj, R. Realizing drug repositioning by adapting a recommendation system to handle the process. *BMC Bioinformatics*, **19** (2018), 136.

[178] Aliper, A., Plis, S., Artemov, A., Ulloa, A., Mamoshina, P., and Zhavoronkov, A. Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Molecular Pharmaceutics*, **13** (2016), 2524–2530.

[179] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, **25** (2012).

[180] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, **9** (1997), 1735–1780.

[181] Altae-Tran, H., Ramsundar, B., Pappu, A. S., and Pande, V. Low data drug discovery with one-shot learning. *ACS Central Science*, **3** (2017), 283–293.

[182] Hu, S., Zhang, C., Chen, P., Gu, P., Zhang, J., and Wang, B. Predicting drug-target interactions from drug structure and protein sequence using novel convolutional neural networks. *BMC Bioinformatics*, **20** (2019), 689.

[183] Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings* (edited by Y. Bengio and Y. LeCun) (2014).

[184] Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics*, **35** (2019), 5191–5198.

[185] Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, **34** (2018), i457–i466.

[186] Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings* (2017).

[187] Doshi, S. and Chepuri, S. P. A computational approach to drug repurposing using graph neural networks. *Computers in Biology and Medicine*, **150** (2022), 105992.

[188] Chakraborty, C., Sharma, A. R., Bhattacharya, M., Agoramoorthy, G., and Lee, S.-S. The drug repurposing for COVID-19 clinical trials provide very effective therapeutic combinations: lessons learned from major clinical studies. *Frontiers in Pharmacology*, **12** (2021), 704205.

[189] Rodrigues, L., Bento Cunha, R., Vassilevskaia, T., Viveiros, M., and Cunha, C. Drug repurposing for COVID-19: a review and a novel strategy to identify new targets and potential drug candidates. *Molecules*, **27** (2022), 2723.

[190] Stolfi, P., Manni, L., Soligo, M., Vergni, D., and Tieri, P. Designing a network proximity-based drug repurposing strategy for COVID-19. *Frontiers in Cell and Developmental Biology*, **8** (2020), 545089.

[191] Smith, D. P., Oechsle, O., Rawling, M. J., Savory, E., Lacoste, A., and Richardson, P. J. Expert-augmented computational drug repurposing identified baricitinib as a treatment for COVID-19. *Frontiers in Pharmacology*, **12** (2021), 709856.

[192] Galindez, G., Matschinske, J., Rose, T. D., Sadegh, S., Salgado-Albarrán, M., Späth, J., Baumbach, J., and Pauling, J. K. Lessons from the COVID-19 pandemic for advancing computational drug repurposing strategies. *Nature Computational Science*, **1** (2021), 33–41.

[193] Mohamed, K., Yazdanpanah, N., Saghazadeh, A., and Rezaei, N. Computational drug discovery and repurposing for the treatment of COVID-19: a systematic review. *Bioorganic chemistry*, **106** (2021), 104490.

[194] Castelvecchi, D. Can we open the black box of AI? *Nature News*, **538** (2016), 20.

[195] Rodríguez-Pérez, R. and Bajorath, J. Chemistry-centric explanation of machine learning models. *Artificial Intelligence in the Life Sciences*, **1** (2021), 100009.

[196] Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, **1** (2019), 206–215.

[197] Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A., and Telenti, A. A primer on deep learning in genomics. *Nature Genetics*, **51** (2019), 12–18.

[198] Belle, V. and Papantonis, I. Principles and practice of explainable machine learning. *Frontiers in Big Data*, **4** (2021), 688969.

[199] Gunning, D., Vorm, E., Wang, Y., and Turek, M. DARPA's explainable AI (XAI) program: a retrospective. *Authorea Preprints*, (2021).

[200] Jiménez-Luna, J., Grisoni, F., and Schneider, G. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, **2** (2020), 573–584.

[201] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., and Zhu, J. Explainable AI: a brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Proceedings, Part II 8*, p. 563–574. Springer (2019).

[202] Feng, J., Lansford, J. L., Katsoulakis, M. A., and Vlachos, D. G. Explainable and trustworthy artificial intelligence for correctable modeling in chemical sciences. *Science Advances*, **6** (2020), eabc3204.

[203] Letzgus, S., Wagner, P., Lederer, J., Samek, W., Müller, K.-R., and Montavon, G. Toward explainable artificial intelligence for regression models: a methodological perspective. *IEEE Signal Processing Magazine*, **39** (2022), 40–58.

[204] Rodríguez-Pérez, R. and Bajorath, J. Interpretation of machine learning models using Shapley values: application to compound potency and multi-target activity predictions. *Journal of Computer-Aided Molecular Design*, **34** (2020), 1013–1026.

[205] Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, p. 3145–3153. PMLR (2017).

[206] Beckers, S. Causal explanations and XAI. In *Conference on Causal Learning and Reasoning*, p. 90–109. PMLR (2022).

[207] Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL & Tech.*, **31** (2017), 841.

[208] Jacovi, A., Swayamdipta, S., Ravfogel, S., Elazar, Y., Choi, Y., and Goldberg, Y. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 1597–1611. Association for Computational Linguistics (2021).

[209] Garson, G. D. Interpreting neural-network connection weights. *AI Expert*, **6** (1991), 46–51.

[210] Goh, A. T. Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering*, **9** (1995), 143–151.

[211] Maozhun, S. and Ji, L. Improved garson algorithm based on neural network model. In *2017 29th Chinese Control And Decision Conference (CCDC)*, p. 4307–4312. IEEE (2017).

[212] Özesmi, S. L. and Özesmi, U. An artificial neural network approach to spatial habitat modelling with interspecific interaction. *Ecological Modelling*, **116** (1999), 15–31.

[213] Olden, J. D., Joy, M. K., and Death, R. G. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, **178** (2004), 389–397.

[214] Dimopoulos, Y., Bourret, P., and Lek, S. Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural Processing Letters*, **2** (1995), 1–4.

[215] Dimopoulos, I., Chronopoulos, J., Chronopoulou-Sereli, A., and Lek, S. Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in Athens city (Greece). *Ecological Modelling*, **120** (1999), 157–165.

[216] Hechtlinger, Y. Interpretation of prediction models using the input gradient. In *30th Conference on Neural Information Processing Systems, NIPS 2016, Workshop on Interpretable Machine Learning in Complex Systems* (2016).

[217] Ross, A. S., Hughes, M. C., and Doshi-Velez, F. Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, p. 2662–2670 (2017).

[218] Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, p. 3319–3328. PMLR (2017).

[219] Yosinski, J., Clune, J., Nguyen, A. M., Fuchs, T. J., and Lipson, H. Understanding neural networks through deep visualization. *CoRR*, **abs/1506.06579** (2015). `arXiv:1506.06579`.

[220] Simonyan, K., Vedaldi, A., and 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* (edited by Y. Bengio and Y. LeCun) (2014).

[221] Murdoch, W. J., Liu, P. J., and Yu, B. Beyond word importance: contextual decomposition to extract interactions from lstms. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings* (2018).

[222] Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1135–1144 (2016).

[223] Tsang, M., Cheng, D., and Liu, Y. Detecting statistical interactions from neural network weights. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings* (2018).

[224] Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., and Tang, J. AutoInt: automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, p. 1161–1170 (2019).

[225] Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. GNNExplainer: generating explanations for graph neural networks. *Advances in Neural Information Processing Systems*, **32** (2019).

[226] Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. Parameterized explainer for graph neural network. *Advances in Neural Information Processing Systems*, **33** (2020).

[227] Gao, Y., Sun, T., Bhatt, R., Yu, D., Hong, S., and Zhao, L. GNES: learning to explain graph neural networks. In *2021 IEEE International Conference on Data Mining (ICDM)*, p. 131–140. IEEE (2021).

[228] Kasanishi, T., Wang, X., and Yamasaki, T. Edge-level explanations for graph neural networks by extending explainability methods for convolutional neural networks. In *2021 IEEE International Symposium on Multimedia (ISM)*, p. 249–252. IEEE (2021).

[229] Xiong, Z., et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, **63** (2019), 8749–8760.

[230] Tang, B., Kramer, S. T., Fang, M., Qiu, Y., Wu, Z., and Xu, D. A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *Journal of Cheminformatics*, **12** (2020), 15.

[231] Dai, E. and Wang, S. Towards self-explainable graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, p. 302–311 (2021).

[232] Huang, Q., Yamada, M., Tian, Y., Singh, D., and Chang, Y. GraphLIME: local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, **35** (2022), 6968.

[233] Wang, T., Dai, X., and Liu, Y. Learning with Hilbert–Schmidt independence criterion: a review and new perspectives. *Knowledge-based Systems*, **234** (2021), 107567.

[234] Vu, M. and Thai, M. T. PGM-Explainer: probabilistic graphical model explanations for graph neural networks. *Advances in Neural Information Processing Systems*, **33** (2020).

[235] Yuan, H., Tang, J., Hu, X., and Ji, S. XGNN: towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 430–438 (2020).

[236] Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, **41** (2014), 647–665.

[237] Young, H. P. Monotonic solutions of cooperative games. *International Journal of Game Theory*, **14** (1985), 65–72.

[238] Lundberg, S. M., et al. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, **2** (2020), 56–67.

[239] Rodríguez-Pérez, R. and Bajorath, J. Interpretation of compound activity predictions from complex machine learning models using local approximations and Shapley values. *Journal of Medicinal Chemistry*, **63** (2019), 8761–8777.

[240] Duval, A. and Malliaros, F. D. GraphSVX: Shapley value explanations for graph neural networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Proceedings, Part II 21*, p. 302–318. Springer (2021).

[241] Yuan, H., Yu, H., Wang, J., Li, K., and Ji, S. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*, p. 12241–12252. PMLR (2021).

[242] Perotti, A., Bajardi, P., Bonchi, F., and Panisson, A. Explaining identity-aware graph classifiers through the language of motifs. In *2023 International Joint Conference on Neural Networks (IJCNN)*, p. 1–8. IEEE (2023).

[243] Gutiérrez-Gómez, L. and Delvenne, J.-C. Unsupervised network embeddings with node identity awareness. *Applied Network Science*, **4** (2019), 82.

[244] Opap, K. and Mulder, N. Recent advances in predicting gene–disease associations. *F1000Research*, **6** (2017), 578.

[245] Piro, R. M. and Cunto, F. D. Computational approaches to disease-gene prediction: rationale, classification and successes. *FEBS Journal*, **279** (2012), 678–696.

[246] Liu, B., Dai, Y., Li, X., Lee, W. S., and Yu, P. S. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, p. 179–186 (2003).

[247] Wang, L., Han, M., Li, X., Zhang, N., and Cheng, H. Review of classification methods on unbalanced data sets. *IEEE Access*, **9** (2021), 64606–64628.

[248] Elkan, C. and Noto, K. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 213–220 (2008).

[249] Claesen, M., De Smet, F., Suykens, J. A., and De Moor, B. A robust ensemble approach to learn from positive and unlabeled data using SVM base models. *Neurocomputing*, **160** (2015), 73–84.

[250] Ke, T., Lv, H., Sun, M., and Zhang, L. A biased least squares support vector machine based on Mahalanobis distance for PU learning. *Physica A: Statistical Mechanics and its Applications*, **509** (2018), 422–438.

[251] Yang, P., Li, X.-L., Mei, J.-P., Kwoh, C.-K., and Ng, S.-K. Positive-unlabeled learning for disease gene identification. *Bioinformatics*, **28** (2012), 2640–2647.

[252] Yang, P., Li, X., Chua, H.-N., Kwoh, C.-K., and Ng, S.-K. Ensemble positive unlabeled learning for disease gene identification. *PLOS ONE*, **9** (2014), e97079.

[253] Can, T., Çamoğlu, O., and Singh, A. K. Analysis of protein-protein interaction networks using random walks. In *Proceedings of the 5th international workshop on Bioinformatics*, p. 61–68 (2005).

[254] Köhler, S., Bauer, S., Horn, D., and Robinson, P. N. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, **82** (2008), 949–958.

[255] Li, Y. and Patra, J. C. Integration of multiple data sources to prioritize candidate genes using discounted rating system. *BMC Bioinformatics*, **11** (2010), S20.

[256] Li, Y. and Patra, J. C. Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics*, **26** (2010), 1219–1224.

[257] Doncheva, N. T., Kacprowski, T., and Albrecht, M. Recent approaches to the prioritization of candidate disease genes. *WIREs Systems Biology and Medicine*, **4** (2012), 429–442.

[258] Silverman, E. K., et al. Molecular networks in network medicine: development and applications. *WIREs Systems Biology and Medicine*, **12** (2020), e1489.

[259] Tieri, P., Farina, L., Petti, M., Astolfi, L., Paci, P., Castiglione, F., et al. Network inference and reconstruction in bioinformatics. In *Encyclopedia of Bioinformatics and Computational Biology*, p. 805–813. Elsevier (2019).

[260] Petti, M., Farina, L., Francone, F., Lucidi, S., Macali, A., Palagi, L., and De Santis, M. MOSES: a new approach to integrate interactome topology and functional features for disease gene prediction. *Genes*, **12** (2021), 1713.

[261] Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, **34** (2006), D535–D539.

[262] Piñero, J., Queralt-Rosinach, N., Bravo, A., Deu-Pons, J., Bauer-Mehren, A., Baron, M., Sanz, F., and Furlong, L. I. DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database*, **2015** (2015), bav028.

[263] The UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Research*, **43** (2015), D204–D212.

[264] Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., McMorran, R., Wiegers, J., Wiegers, T. C., and Mattingly, C. J. The comparative toxicogenomics database: update 2019. *Nucleic Acids Research*, **47** (2019), D948–D954.

[265] Rehm, H. L., et al. ClinGen — the clinical genome resource. *New England Journal of Medicine*, **372** (2015), 2235–2242.

[266] Martin, A. R., et al. PanelApp crowdsources expert knowledge to establish consensus diagnostic gene panels. *Nature Genetics*, **51** (2019), 1560–1565.

[267] Tamborero, D., et al. Cancer genome interpreter annotates the biological and clinical relevance of tumor alterations. *Genome Medicine*, **10** (2018), 25.

[268] Gutiérrez-Sacristán, A., Grosdidier, S., Valverde, O., Torrens, M., Bravo, À., Piñero, J., Sanz, F., and Furlong, L. I. PsyGeNET: a knowledge platform on psychiatric disorders and their genes. *Bioinformatics*, **31** (2015), 3075–3077.

[269] Bundschus, M., Dejori, M., Stetter, M., Tresp, V., and Kriegel, H.-P. Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, **9** (2008), 207.

[270] Bundschus, M., Bauer-Mehren, A., Tresp, V., Furlong, L., and Kriegel, H.-P. Digging for knowledge with information extraction: a case study on human gene-disease associations. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, p. 1845–1848 (2010).

[271] Bravo, A., Cases, M., Queralt-Rosinach, N., Sanz, F., and Furlong, L. A knowledge-driven approach to extract disease-related biomarkers from the literature. *BioMed Research International*, **2014** (2014), 253128.

[272] Bravo, À., Piñero, J., Queralt-Rosinach, N., Rautschka, M., and Furlong, L. I. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC Bioinformatics*, **16** (2015), 55.

[273] Lancour, D., Naj, A., Mayeux, R., Haines, J. L., Pericak-Vance, M. A., Schellenberg, G. D., Crovella, M., Farrer, L. A., and Kasif, S. One for all and all for one: improving replication of genetic studies through network diffusion. *PLoS Genetics*, **14** (2018), e1007306.

[274] Picart-Armada, S., Barrett, S. J., Willé, D. R., Perera-Lluna, A., Gutteridge, A., and Dessailly, B. H. Benchmarking network propagation methods for disease gene identification. *PLoS Computational Biology*, **15** (2019), e1007276.

[275] Janyasupab, P., Suratanee, A., and Plaimas, K. Network diffusion with centrality measures to identify disease-related genes. *Mathematical Biosciences and Engineering*, **18** (2021), 2909–2929.

[276] Carlin, D. E., Demchak, B., Pratt, D., Sage, E., and Ideker, T. Network propagation in the Cytoscape cyberinfrastructure. *PLoS Computational Biology*, **13** (2017), e1005598.

[277] Nitsch, D., Gonçalves, J. P., Ojeda, F., De Moor, B., and Moreau, Y. Candidate gene prioritization by network analysis of differential expression using machine learning approaches. *BMC Bioinformatics*, **11** (2010), 460.

[278] White, S. and Smyth, P. Algorithms for estimating relative importance in networks. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 266–275 (2003).

[279] Baronchelli, A. and Loreto, V. Ring structures and mean first passage time in networks. *Physical Review E*, **73** (2006), 026103.

[280] Xu, J. and Li, Y. Discovering disease-genes by topological features in human protein–protein interaction network. *Bioinformatics*, **22** (2006), 2800–2805.

[281] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C.-F. A new method to measure the semantic similarity of GO terms. *Bioinformatics*, **23** (2007), 1274–1281.

[282] Ashburner, M., et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, **25** (2000), 25–29.

[283] Enright, A. J., Van Dongen, S., and Ouzounis, C. A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, **30** (2002), 1575–1584.

[284] Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., and Singh, M. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, **21** (2005), i302–i310.

[285] Chen, E. Y., Tan, C. M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G. V., Clark, N. R., and Ma'ayan, A. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics*, **14** (2013), 128.

[286] Kuleshov, M. V., et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Research*, **44** (2016), W90–W97.

[287] Xie, Z., et al. Gene set knowledge discovery with Enrichr. *Current Protocols*, **1** (2021), e90.

[288] Chen, X., et al. Apoptotic engulfment pathway and schizophrenia. *PLOS ONE*, **4** (2009), e6875.

[289] Scaini, G., Fries, G. R., Valvassori, S. S., Zeni, C. P., Zunta-Soares, G., Berk, M., Soares, J. C., and Quevedo, J. Perturbations in the

apoptotic pathway and mitochondrial network dynamics in peripheral blood mononuclear cells from bipolar disorder patients. *Translational Psychiatry*, **7** (2017), e1111.

[290] Valentini, C. G., Fianchi, L., Voso, M. T., Caira, M., Leone, G., and Pagano, L. Incidence of acute myeloid leukemia after breast cancer. *Mediterranean Journal of Hematology and Infectious Diseases*, **3** (2011), e2011069.

[291] Fernández, L. P., et al. The role of glycosyltransferase enzyme GCNT3 in colon and ovarian cancer prognosis and chemoresistance. *Scientific Reports*, **8** (2018), 8485.

[292] Cummings, J. L. Depression and Parkinson's disease: a review. *American Journal Psychiatry*, **149** (1992), 443–454.

[293] Shelton, R. C., Claiborne, J., Sidoryk-Wegrzynowicz, M., Reddy, R., Aschner, M., Lewis, D. A., and Mirnics, K. Altered expression of genes involved in inflammation and apoptosis in frontal cortex in major depression. *Molecular Psychiatry*, **16** (2011), 751–762.

[294] Mehkari, Z., Mohammed, L., Javed, M., Althwanay, A., Ahsan, F., Oliveri, F., Goud, H. K., and Rutkofsky, I. H. Manganese, a Likely Cause of 'Parkinson's in Cirrhosis', a Unique Clinical Entity of Acquired Hepatocerebral Degeneration. *Cureus*, **12** (2020), e10448.

[295] Cole-Clark, D., Nair-Shalliker, V., Bang, A., Rasiah, K., Chalasani, V., and Smith, D. P. An initial melanoma diagnosis may increase the subsequent risk of prostate cancer: results from the New South Wales Cancer Registry. *Scientific Reports*, **8** (2018), 7167.

[296] Zigman, W. B. and Lott, I. T. Alzheimer's disease in Down syndrome: neurobiology and risk. *Mental Retardation and Developmental Disabilities Research Reviews*, **13** (2007), 237–246.

[297] Yan-Hong, H., Jing, L., Hong, L., Shan-Shan, H., Yan, L., and Ju, L. Association between alcohol consumption and the risk of ovarian cancer: a meta-analysis of prospective observational studies. *BMC Public Health*, **15** (2015), 223.

[298] Erol, A., Ho, A. M.-C., Winham, S. J., and Karpyak, V. M. Sex hormones in alcohol consumption: a systematic review of evidence. *Addiction Biology*, **24** (2019), 157–169.

[299] Lee, I.-C. and Chiang, K.-L. Clinical diagnosis and treatment of Leigh Syndrome based on SURF1: genotype and phenotype. *Antioxidants*, **10** (2021), 1950.

[300] de Magalhães, J. P. Every gene can (and possibly will) be associated with cancer. *Trends in Genetics*, **38** (2022), 216–217.

[301] Lazareva, O., Baumbach, J., List, M., and Blumenthal, D. B. On the limits of active module identification. *Briefings in Bioinformatics*, **22** (2021), bbab066.

[302] Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, **30** (2017).

[303] Pfeifer, B., Saranti, A., and Holzinger, A. GNN-SubNet: disease subnetwork detection with explainable graph neural networks. *Bioinformatics*, **38** (2022), ii120–ii126.

[304] Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).

[305] Fukushima, K. Cognitron: a self-organizing multilayered neural network. *Biological Cybernetics*, **20** (1975), 121–136.

[306] Zhao, L. and Akoglu, L. PairNorm: tackling oversmoothing in GNNs. In *8th International Conference on Learning Representations, ICLR 2020* (2020).

[307] Kingma, D. P. and Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* (edited by Y. Bengio and Y. LeCun) (2015).

[308] Menche, J., Sharma, A., Kitsak, M., Ghiassian, S. D., Vidal, M., Loscalzo, J., and Barabási, A.-L. Uncovering disease-disease relationships through the incomplete interactome. *Science*, **347** (2015), 1257601.

[309] Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., and McKusick, V. A. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, **33** (2005), D514–D517.

[310] Ramos, E. M., Hoffman, D., Junkins, H. A., Maglott, D., Phan, L., Sherry, S. T., Feolo, M., and Hindorff, L. A. Phenotype–genotype

integrator (PheGenI): synthesizing genome-wide association study (GWAS) data with existing genomic resources. *European Journal of Human Genetics*, **22** (2014), 144–147.

[311] De Luca, R., Carfora, M., Blanco, G., Mastropietro, A., Petti, M., and Tieri, P. PROCONSUL: probabilistic exploration of connectivity significance patterns for disease module discovery. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 1941–1947. IEEE (2022).

[312] Gentili, M., Martini, L., Sponziello, M., and Becchetti, L. Biological random walks: multi-omics integration for disease gene prioritization. *Bioinformatics*, **38** (2022), 4145–4152.

[313] Jin, H. and Zangar, R. C. Protein modifications as potential biomarkers in breast cancer. *Biomarker Insights*, **4** (2009), 191–200.

[314] Bradner, J. E., Hnisz, D., and Young, R. A. Transcriptional addiction in cancer. *Cell*, **168** (2017), 629–643.

[315] Reed, J. C. Apoptosis-targeted therapies for cancer. *Cancer Cell*, **3** (2003), 17–22.

[316] Plati, J., Bucur, O., and Khosravi-Far, R. Apoptotic cell signaling in cancer progression and therapy. *Integrative Biology*, **3** (2011), 279–296.

[317] Pfeffer, C. M. and Singh, A. T. Apoptosis: a target for anticancer therapy. *International Journal of Molecular Sciences*, **19** (2018), 448.

[318] Owens, S. J., Weickert, T. W., Purves-Tyson, T. D., Ji, E., White, C., Galletly, C., Liu, D., O'Donnell, M., and Weickert, C. S. Sex-specific associations of androgen receptor CAG trinucleotide repeat length and of raloxifene treatment with testosterone levels and perceived stress in schizophrenia. *Complex Psychiatry*, **5** (2019), 28–41.

[319] Wang, K. Molecular mechanisms of hepatic apoptosis. *Cell Death & Disease*, **5** (2014), e996–e996.

[320] Guicciardi, M. and Gores, G. Apoptosis: a mechanism of acute and chronic liver injury. *Gut*, **54** (2005), 1024–1033.

[321] Schmidt, S., Denk, S., and Wiegering, A. Targeting protein synthesis in colorectal cancer. *Cancers*, **12** (2020), 1298.

[322] McKenzie, S. and Kyprianou, N. Apoptosis evasion: the role of survival pathways in prostate cancer progression and therapeutic resistance. *Journal of Cellular Biochemistry*, **97** (2006), 18–32.

[323] Ali, A. and Kulik, G. Signaling pathways that control apoptosis in prostate cancer. *Cancers*, **13** (2021), 937.

[324] Turner, M. R., Goldacre, R., Talbot, K., and Goldacre, M. J. Psychiatric disorders prior to amyotrophic lateral sclerosis. *Annals of Neurology*, **80** (2016), 935–938.

[325] Au, P., et al. Phenotypic spectrum of Au–Kline syndrome: a report of six new cases and review of the literature. *European Journal of Human Genetics*, **26** (2018), 1272–1281.

[326] Wang, K., Zhang, S., Marzolf, B., Troisch, P., Brightman, A., Hu, Z., Hood, L. E., and Galas, D. J. Circulating microRNAs, potential biomarkers for drug-induced liver injury. *Proceedings of the National Academy of Sciences*, **106** (2009), 4402–4407.

[327] Shelton, R., Claiborne, J., Sidoryk-Wegrzynowicz, M., Reddy, R., Aschner, M., Lewis, D., and Mirnics, K. Altered expression of genes involved in inflammation and apoptosis in frontal cortex in major depression. *Molecular Psychiatry*, **16** (2011), 751–762.

[328] Sachdeva, A., Chandra, M., Choudhary, M., Dayal, P., and Anand, K. S. Alcohol-related dementia and neurocognitive impairment: a review study. *International Journal of High Risk Behaviors & Addiction*, **5** (2016), e27976.

[329] Kamburov, A., Wierling, C., Lehrach, H., and Herwig, R. ConsensusPathDB — a database for integrating human functional interaction networks. *Nucleic Acids Research*, **37** (2009), D623–D628.

[330] Kamburov, A., Stelzl, U., Lehrach, H., and Herwig, R. The ConsensusPathDB interaction database: 2013 update. *Nucleic Acids Research*, **41** (2013), D793–D800.

[331] Kamburov, A. and Herwig, R. ConsensusPathDB 2022: molecular interactions update as a resource for network biology. *Nucleic Acids Research*, **50** (2022), D587–D595.

[332] Evangelou, E., et al. Genetic analysis of over 1 million people identifies 535 new loci associated with blood pressure traits. *Nature Genetics*, **50** (2018), 1412–1425.

[333] Sudlow, C., et al. UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine*, **12** (2015), e1001779.

[334] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* (edited by Y. Bengio and Y. LeCun) (2015).

[335] Pruitt, K. D., et al. The consensus coding sequence (CCDS) project: identifying a common protein-coding gene set for the human and mouse genomes. *Genome Research*, **19** (2009), 1316–1323.

[336] Harrow, J., et al. GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Research*, **22** (2012), 1760–1774.

[337] Sherry, S. T., Ward, M.-H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., and Sirotkin, K. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, **29** (2001), 308–311.

[338] Birney, E., et al. An overview of ensembl. *Genome Research*, **14** (2004), 925–928.

[339] Cordell, H. J. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, **10** (2009), 392–404.

[340] Guo, X., Meng, Y., Yu, N., and Pan, Y. Cloud computing for detecting high-order genome-wide epistatic interaction via dynamic clustering. *BMC Bioinformatics*, **15** (2014), 102.

[341] Ma'ayan, A. Introduction to network analysis in systems biology. *Science Signaling*, **4** (2011), tr5.

[342] Carbon, S., Ireland, A., Mungall, C. J., Shu, S., Marshall, B., Lewis, S., Hub, A., and Group, W. P. W. AmiGO: online access to ontology and annotation data. *Bioinformatics*, **25** (2009), 288–289.

[343] Sollis, E., et al. The NHGRI-EBI GWAS Catalog: knowledgebase and deposition resource. *Nucleic Acids Research*, **51** (2023), D977–D985.

[344] Dono, R., Texido, G., Dussel, R., Ehmke, H., and Zeller, R. Impaired cerebral cortex development and blood pressure regulation in FGF-2-deficient mice. *The EMBO Journal*, **17** (1998), 4213–4225.

[345] Carmichael, C. Y. and Wainford, R. D. Hypothalamic signaling mechanisms in hypertension. *Current Hypertension Reports*, **17** (2015), 39.

[346] Ahmed, A. A., Mooar, P. A., Kleiner, M., Torg, J. S., and Miyamoto, C. T. Hypertensive patients show delayed wound healing following total hip arthroplasty. *PLOS ONE*, **6** (2011), e23224.

[347] Rinaldi, G. and Bohr, D. Plasma membrane and its abnormalities in hypertension. *The American Journal of the Medical Sciences*, **295** (1988), 389–395.

[348] de Leeuw, C. A., Stringer, S., Dekkers, I. A., Heskes, T., and Posthuma, D. Conditional and interaction gene-set analysis reveals novel functional pathways for blood pressure. *Nature Communications*, **9** (2018), 3768.

[349] Heffernan, K. S., Maron, M. S., Patvardhan, E. A., Karas, R. H., Kuvin, J. T., Group, V. F. S., et al. Relation of pulse pressure to blood pressure response to exercise in patients with hypertrophic cardiomyopathy. *The American Journal of Cardiology*, **107** (2011), 600–603.

[350] Taylor, M. B. and Ehrenreich, I. M. Higher-order genetic interactions and their contribution to complex traits. *Trends in Genetics*, **31** (2015), 34–40.

[351] Greenside, P., Shimko, T., Fordyce, P., and Kundaje, A. Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. *Bioinformatics*, **34** (2018), i629–i637.

[352] Moore, J. H. and Williams, S. M. Traversing the conceptual divide between biological and statistical epistasis: systems biology and a more modern synthesis. *Bioessays*, **27** (2005), 637–646.

[353] Gjuvsland, A. B., Hayes, B. J., Omholt, S. W., and Carlborg, O. Statistical epistasis is a generic feature of gene regulatory networks. *Genetics*, **175** (2007), 411–420.

[354] Aylor, D. L. and Zeng, Z.-B. From classical genetics to quantitative genetics to systems biology: modeling epistasis. *PLoS Genetics*, **4** (2008), e1000029.

[355] Phillips, P. C. Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, **9** (2008), 855–867.

[356] Koschützki, D. and Schreiber, F. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene Regulation and Systems Biology*, **2** (2008), 193–201.

[357] Lin, C.-Y., Chin, C.-H., Wu, H.-H., Chen, S.-H., Ho, C.-W., and Ko, M.-T. Hubba: hub objects analyzer—a framework of interactome hubs identification for network biology. *Nucleic Acids Research*, **36** (2008), W438–W443.

[358] Wang, N., Xu, H.-l., Zhao, X., Wen, X., Wang, F.-t., Wang, S.-y., Fu, L.-l., Liu, B., and Bao, J.-k. Network-based identification of novel connections among apoptotic signaling pathways in cancer. *Applied Biochemistry and Biotechnology*, **167** (2012), 621–631.

[359] Deng, S.-P., Zhu, L., and Huang, D.-S. Predicting hub genes associated with cervical cancer through gene co-expression networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **13** (2015), 27–35.

[360] Tang, J., Kong, D., Cui, Q., Wang, K., Zhang, D., Gong, Y., and Wu, G. Prognostic genes of breast cancer identified by gene co-expression network analysis. *Frontiers in Oncology*, **8** (2018), 374.

[361] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, **13** (2003), 2498–2504.

[362] Callera, G. E., Montezano, A. C., Yogi, A., Tostes, R. C., He, Y., Schiffrin, E. L., and Touyz, R. M. c-Src–dependent nongenomic signaling responses to aldosterone are increased in vascular myocytes from spontaneously hypertensive rats. *Hypertension*, **46** (2005), 1032–1038.

[363] Bao, W., et al. Effects of p38 MAPK inhibitor on angiotensin II-dependent hypertension, organ damage, and superoxide anion production. *Journal of Cardiovascular Pharmacology*, **49** (2007), 362–368.

[364] Beuschlein, F., et al. Somatic mutations in ATP1A1 and ATP2B3 lead to aldosterone-producing adenomas and secondary hypertension. *Nature Genetics*, **45** (2013), 440–444.

[365] Schiffl, H., Küchle, C., and Lang, S. Dietary salt, intracellular ion homeostasis and hypertension secondary to early-stage kidney disease. *Mineral and Electrolyte Metabolism*, **22** (1996), 178–181.

[366] Sabater-Molina, M., Pérez-Sánchez, I., Hernández Del Rincón, J., and Gimeno, J. Genetics of hypertrophic cardiomyopathy: a review of current state. *Clinical Genetics*, **93** (2018), 3–14.

[367] Hurst, L. A., et al. TNFα drives pulmonary arterial hypertension by suppressing the BMP type-II receptor and altering NOTCH signalling. *Nature Communications*, **8** (2017), 14079.

[368] Smith, K. A., et al. Notch activation of Ca2+ signaling in the development of hypoxic pulmonary vasoconstriction and pulmonary hypertension. *American Journal of Respiratory Cell and Molecular Biology*, **53** (2015), 355–367.

[369] Hirschfield, G. M., Beuers, U., Corpechot, C., Invernizzi, P., Jones, D., Marzioni, M., and Schramm, C. EASL clinical practice guidelines: the diagnosis and management of patients with primary biliary cholangitis. *Journal of Hepatology*, **67** (2017), 145–172.

[370] Shahini, E. and Ahmed, F. Chronic fatigue should not be overlooked in primary biliary cholangitis. *Journal of Hepatology*, **75** (2021), 744–745.

[371] Bowlus, C. L., et al. Long-term obeticholic acid therapy improves histological endpoints in patients with primary biliary cholangitis. *Clinical Gastroenterology and Hepatology*, **18** (2020), 1170–1178.

[372] Kjærgaard, K., et al. Obeticholic acid improves hepatic bile acid excretion in patients with primary biliary cholangitis. *Journal of Hepatology*, **74** (2021), 58–65.

[373] Alvaro, D., Carpino, G., Craxi, A., Floreani, A., Moschetta, A., and Invernizzi, P. Primary biliary cholangitis management: controversies, perspectives and daily practice implications from an expert panel. *Liver International*, **40** (2020), 2590–2601.

[374] Kowdley, K. V., et al. A randomized trial of obeticholic acid monotherapy in patients with primary biliary cholangitis. *Hepatology*, **67** (2018), 1890–1902.

[375] Karlsen, T., Vesterhus, M., and Boberg, K. Controversies in the management of primary biliary cirrhosis and primary sclerosing cholangitis. *Alimentary Pharmacology & Therapeutics*, **39** (2014), 282–301.

[376] Hirschfield, G. M. and Gershwin, M. E. The immunobiology and pathophysiology of primary biliary cirrhosis. *Annual Review of Pathology: Mechanisms of Disease*, **8** (2013), 303–330.

[377] Lindor, K. D., Gershwin, E. M., Poupon, R., Kaplan, M., Bergasa, N. V., and Heathcote, J. E. Primary biliary cirrhosis. *Hepatology*, **50** (2009), 291–308.

[378] European Association For The Study Of The Liver. EASL clinical practice guidelines: management of cholestatic liver diseases. *Journal of Hepatology*, **51** (2009), 237–267.

[379] Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., and Woolsey, J. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, **34** (2006), D668–D672.

[380] Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., and Hassanali, M. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, **36** (2008), D901–D906.

[381] Povey, S., Lovering, R., Bruford, E., Wright, M., Lush, M., and Wain, H. The HUGO gene nomenclature committee (HGNC). *Human genetics*, **109** (2001), 678–680.

[382] Wang, J., Vasaikar, S., Shi, Z., Greer, M., and Zhang, B. WebGestalt 2017: a more comprehensive, powerful, flexible and interactive gene set enrichment analysis toolkit. *Nucleic Acids Research*, **45** (2017), W130–W137.

[383] Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, **45** (2017), D353–D361.

[384] Fabregat, A., et al. The Reactome pathway knowledgebase. *Nucleic Acids Research*, **46** (2018), D649–D655.

[385] Jourquin, J., Duncan, D., Shi, Z., and Zhang, B. GLAD4U: deriving and prioritizing gene lists from pubmed literature. *BMC Genomics*, **13** (2012), S20.

[386] Albensi, B. C. What is nuclear factor kappa B (NF-κB) doing in and to the mitochondrion? *Frontiers in Cell and Developmental Biology*, **7** (2019), 154.

[387] Yu, Y., Li, M.-P., Xu, B., Fan, F., Lu, S.-F., Pan, M., and Wu, H.-S. A study of regulatory effects of TLR4 and NF-κB on primary biliary cholangitis. *European Review for Medical & Pharmacological Sciences*, **23** (2019).

[388] Feagan, B. G., et al. Ustekinumab as induction and maintenance therapy for Crohn's disease. *New England Journal of Medicine*, **375** (2016), 1946–1960.

[389] Hirschfield, G. M., et al. Ustekinumab for patients with primary biliary cholangitis who have an inadequate response to ursodeoxycholic acid: a proof-of-concept study. *Hepatology*, **64** (2016), 189–199.

[390] Vastert, S. J., Jamilloux, Y., Quartier, P., Ohlman, S., Osterling Koskinen, L., Kullenberg, T., Franck-Larsson, K., Fautrel, B., and De Benedetti, F. Anakinra in children and adults with still's disease. *Rheumatology*, **58** (2019), vi9–vi22.

[391] Gordon, S. C., Trudeau, S., Regev, A., Uhas, J. M., Chakladar, S., Pinto-Correia, A., Gottlieb, K., and Schlichting, D. Baricitinib and primary biliary cholangitis. *Journal of Translational Autoimmunity*, **4** (2021), 100107.

[392] Crosignani, A., Battezzati, P. M., Setchell, K. D., Invernizzi, P., Covini, G., Zuin, M., and Podda, M. Tauroursodeoxycholic acid for treatment of primary biliary cirrhosis: a dose-response study. *Digestive Diseases and Sciences*, **41** (1996), 809–815.

[393] Larghi, A., Crosignani, A., Battezzati, P., De Valle, G., Allocca, M., Invernizzi, P., Zuin, M., and Podda, M. Ursodeoxycholic and tauro-ursodeoxycholic acids for the treatment of primary biliary cirrhosis: a pilot crossover study. *Alimentary Pharmacology & Therapeutics*, **11** (1997), 409–414.

[394] Ma, H., et al. A multicenter, randomized, double-blind trial comparing the efficacy and safety of TUDCA and UDCA in Chinese patients with primary biliary cholangitis. *Medicine*, **95** (2016), e5391.

[395] Shah, R. A. and Kowdley, K. V. Current and potential treatments for primary biliary cholangitis. *The Lancet Gastroenterology & Hepatology*, **5** (2020), 306–315.

[396] Lleo, A., Ma, X., Gershwin, M. E., and Invernizzi, P. Might denosumab fit in primary biliary cholangitis treatment? *Hepatology*, **72** (2020), 359–360.

[397] Kereiakes, D. J. Inflammation as a therapeutic target: a unique role for abciximab. *American Heart Journal*, **146** (2003), S1–S4.

[398] Wilde, M. I. and Goa, K. L. Muromonab CD3: a reappraisal of its pharmacology and use as prophylaxis of solid organ transplant rejection. *Drugs*, **51** (1996), 865–894.

[399] Ueda, D., Hori, T., Nguyen, J. H., and Uemoto, S. Muromonab-CD3 therapy for refractory rejections after liver transplantation: a single-center experience during two decades in Japan. *Journal of Hepato-Biliary-Pancreatic Sciences*, **17** (2010), 885–891.

[400] Xiao, J., Gao, M., Sun, Z., Diao, Q., Wang, P., and Gao, F. Recent advances of podophyllotoxin/epipodophyllotoxin hybrids in anticancer activity, mode of action, and structure-activity relationship: an update (2010–2020). *European Journal of Medicinal Chemistry*, **208** (2020), 112830.

[401] Patel, M., et al. Etoposide as salvage therapy for cytokine storm due to coronavirus disease 2019. *Chest*, **159** (2021), e7–e11.

[402] Sano, A., et al. The profiling of plasma free amino acids and the relationship between serum albumin and plasma-branched chain amino acids in chronic liver disease: a single-center retrospective study. *Journal of Gastroenterology*, **53** (2018), 978–988.

[403] ter Borg, P. C., Fekkes, D., Vrolijk, J. M., and van Buuren, H. R. The relation between plasma tyrosine concentration and fatigue in primary biliary cirrhosis and primary sclerosing cholangitis. *BMC Gastroenterology*, **5** (2005), 11.

[404] Yang, F., Tang, X., Ding, L., Zhou, Y., Yang, Q., Gong, J., Wang, G., Wang, Z., and Yang, L. Curcumin protects ANIT-induced cholestasis through signaling pathway of FXR-regulated bile acid and inflammation. *Scientific Reports*, **6** (2016), 33052.

[405] Reddy, A., Prince, M., James, O., Jain, S., and Bassendine, M. Tamoxifen: a novel treatment for primary biliary cirrhosis? *Liver International*, **24** (2004), 194–197.

[406] Henze, L., Schwinge, D., and Schramm, C. The effects of androgens on t cells: clues to female predominance in autoimmune liver diseases? *Frontiers in Immunology*, **11** (2020), 1567.

[407] Ortona, E., Pierdominici, M., and Rider, V. Sex hormones and gender differences in immune responses. *Frontiers in Immunology*, **10** (2019), 1076.

[408] Salas, A. L., Ocampo, G., Fariña, G. G., Reyes-Esparza, J., and Rodríguez-Fragoso, L. Genistein decreases liver fibrosis and cholestasis induced by prolonged biliary obstruction in the rat. *Annals of Hepatology*, **6** (2007), 41–47.

[409] Erdös, P. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, **5** (1960), 17–61.

[410] Yuan, H., Yu, H., Gui, S., and Ji, S. Explainability in graph neural networks: a taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45** (2022), 5782–5799.

[411] Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., and Das, P. Explanations based on the missing: towards contrastive explanations with pertinent negatives. *Advances in Neural Information Processing Systems*, **31** (2018).

[412] Artelt, A. and Hammer, B. Efficient computation of contrastive explanations. In *2021 International Joint Conference on Neural Networks (IJCNN)*, p. 1–9. IEEE (2021).

[413] Lipton, P. Contrastive explanation. *Royal Institute of Philosophy Supplements*, **27** (1990), 247–266.

[414] Gaulton, A., et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, **40** (2012), D1100–D1107.

[415] Bento, A. P., et al. The ChEMBL bioactivity database: an update. *Nucleic Acids Research*, **42** (2014), D1083–D1090.

[416] Mendez, D., et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, **47** (2019), D930–D940.

[417] Bruns, R. F. and Watson, I. A. Rules for identifying potentially reactive or promiscuous compounds. *Journal of Medicinal Chemistry*, **55** (2012), 9763–9772.

[418] Irwin, J. J., Duan, D., Torosyan, H., Doak, A. K., Ziebart, K. T., Sterling, T., Tumanian, G., and Shoichet, B. K. An aggregation advisor for ligand discovery. *Journal of Medicinal Chemistry*, **58** (2015), 7076–7087.

[419] Landrum, G. et al. RDKit: open-source cheminformatics software. `https://www.rdkit.org` (2016).

[420] Kojima, R., Ishida, S., Ohta, M., Iwata, H., Honma, T., and Okuno, Y. kGCN: a graph-based deep learning framework for chemical structures. *Journal of Cheminformatics*, **12** (2020), 32.

[421] Paszke, A., et al. Pytorch: an imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, **32** (2019).

[422] Pedregosa, F., et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, **12** (2011), 2825–2830.

[423] Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, **50** (2010), 742–754.

[424] Forthofer, R. N., Lehnen, R. G., Forthofer, R. N., and Lehnen, R. G. Rank correlation methods. *Public Program Analysis: A New Categorical Data Approach*, (1981), 146–163.

[425] Jiménez-Luna, J., Skalic, M., and Weskamp, N. Benchmarking molecular feature attribution methods with activity cliffs. *Journal of Chemical Information and Modeling*, **62** (2022), 274–283.

[426] Yang, K., et al. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, **59** (2019), 3370–3388.

[427] Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D., and Venkatesh, S. GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, **37** (2021), 1140–1147.

[428] Wang, J. and Dokholyan, N. V. Yuel: improving the generalizability of structure-free compound–protein interaction prediction. *Journal of Chemical Information and Modeling*, **62** (2022), 463–471.

[429] Yang, J., Shen, C., and Huang, N. Predicting or pretending: artificial intelligence for protein-ligand interactions lack of sufficiently large and unbiased datasets. *Frontiers in Pharmacology*, **11** (2020), 69.

[430] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings* (2018).

[431] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019* (2019).

[432] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. S., and Leskovec, J. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020* (2020).

[433] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and Leman go neural: higher-order graph neural networks. In *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 33, p. 4602–4609 (2019).

[434] Wang, R., Fang, X., Lu, Y., Yang, C.-Y., and Wang, S. The PDBbind database: methodologies and updates. *Journal of Medicinal Chemistry*, **48** (2005), 4111–4119.

[435] Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y., and Wang, R. PDB-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, **31** (2015), 405–412.

[436] Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., and Wang, R. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of Chemical Research*, **50** (2017), 302–309.

[437] Schmitt, S., Kuhn, D., and Klebe, G. A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology*, **323** (2002), 387–406.

[438] Desaphy, J., Raimbaud, E., Ducrot, P., and Rognan, D. Encoding protein–ligand interaction patterns in fingerprints and graphs. *Journal of Chemical Information and Modeling*, **53** (2013), 623–637.

[439] Da Silva, F., Desaphy, J., and Rognan, D. IChem: a versatile toolkit for detecting, comparing, and predicting protein–ligand interactions. *ChemMedChem*, **13** (2018), 507–510.

[440] Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference* (edited by G. Varoquaux, T. Vaught, and J. Millman), pp. 11 – 15 (2008).

[441] Ahsan, M. M., Mahmud, M. P., Saha, P. K., Gupta, K. D., and Siddique, Z. Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, **9** (2021), 52.

[442] Johnson, H. M. Clever Hans (the horse of Mr. von Osten): a contribution to experimental, animal, and human psychology by Oskar Pfungst, C. Stumpf, Carl L. Rahn, James R. Angell. *The Journal of Philosophy, Psychology and Scientific Methods*, **8** (1911), 663–666.

[443] Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, **10** (2019), 1096.

[444] Tanimoto, T. T. Elementary mathematical theory of classification and prediction. (1958).

[445] Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. Graph kernels for chemical informatics. *Neural Networks*, **18** (2005), 1093–1110.

[446] Bajorath, J. State-of-the-art of artificial intelligence in medicinal chemistry. *Future Science OA*, **7** (2021), FSO702.

[447] Janzing, D., Minorics, L., and Blöbaum, P. Feature relevance quantification in explainable AI: a causal problem. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020* (edited by S. Chiappa and R. Calandra), vol. 108, p. 2907–2916. PMLR (2020).

[448] Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, p. 144–152 (1992).

[449] Böken, B. On the appropriateness of Platt scaling in classifier calibration. *Information Systems*, **95** (2021), 101641.

[450] Goldt, S., Loureiro, B., Reeves, G., Krzakala, F., Mézard, M., and Zdeborová, L. The Gaussian equivalence of generative models for learning with shallow neural networks. In *Mathematical and Scientific Machine Learning*, p. 426–471. PMLR (2022).

[451] Veiga, R., Stephan, L., Loureiro, B., Krzakala, F., and Zdeborová, L. Phase diagram of stochastic gradient descent in high-dimensional two-layer neural networks. *Advances in Neural Information Processing Systems*, **35** (2022).

[452] Shahini, E., Pasculli, G., Mastropietro, A., Stolfi, P., Tieri, P., Vergni, D., Cozzolongo, R., Giannelli, G., and Pesce, F. Oc. 15.1 network proximity-based drug repurposing strategy for primary biliary cholangitis. *Digestive and Liver Disease*, **54** (2022), S106.

[453] Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature*, **533** (2016), 452–454.

[454] Mastropietro, A., Pasculli, G., and Bajorath, J. Protocol to explain graph neural network predictions using an edge-centric Shapley value-based approach. *STAR Protocols*, **3** (2022), 101887.

[455] Neal, R. M. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, vol. 118 of *Lecture Notes in Statistics*, p. 29–53. Springer (1996).

[456] Daniely, A., Frostig, R., and Singer, Y. Toward deeper understanding of neural networks: the power of initialization and a dual view on expressivity. *Advances in Neural Information Processing Systems*, **29** (2016).

[457] Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as Gaussian processes. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings* (2018).

[458] Yang, G. Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. *Advances in Neural Information Processing Systems*, **32** (2019).

[459] Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, **31** (2018).

[460] Gómez-Bombarelli, R., et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, **4** (2018), 268–276.

[461] Brown, N., Fiscato, M., Segler, M. H., and Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, **59** (2019), 1096–1108.

[462] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings* (2017).

[463] Manica, M., Oskooei, A., Born, J., Subramanian, V., Sáez-Rodríguez, J., and Rodriguez Martinez, M. Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *Molecular Pharmaceutics*, **16** (2019), 4797–4806.