

A Multimodal Graph Convolutional Approach to Predict Genes Associated with Rare Genetic Diseases

Dhruva Sahasrabudhe

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Application

T.M. Murali, Chair

Anuj Karpatne

Bert Huang

August 13, 2020.

Blacksburg, Virginia

Keywords: Graph Machine Learning, Disease Gene Prediction, Graph Convolutional
Networks, Link Prediction

Copyright 2020, Dhruva Sahasrabudhe

A Multimodal Graph Convolutional Approach to Predict Genes Associated with Rare Genetic Diseases

Dhruva Sahasrabudhe

(ABSTRACT)

There exist a large number of rare genetic diseases in humans. Our knowledge of the specific gene variants whose presence in the genome of a person predisposes them towards developing a disease, called gene associations, is incomplete. Computational tools which can predict genes which may be associated with a rare disease have great utility in healthcare. However, a majority of existing prediction algorithms require a set of already known “seed genes” to further discover novel associations for a disease. This drawback becomes more serious for rare genetic diseases, since a large proportion do not have any known gene associations. In this work, we develop an approach for disease-gene association prediction that overcomes the reliance on seed genes. Our approach uses the similarity of the observable biological characteristics of diseases (i.e., phenotypes) along with a global map of direct and indirect human protein interactions, to transfer associations from diseases whose gene associations have been discovered to diseases with no known gene associations. We formulate disease-gene association prediction over a multimodal network of diseases and genes, and develop an approach based on graph convolutional networks. We show how our model design considerations impact prediction performance. We demonstrate that our approach outperforms simpler graph machine learning and traditional machine learning approaches, as well as a competitive network propagation based approach for the task of predicting disease-gene associations.

A Multimodal Graph Convolutional Approach to Predict Genes Associated with Rare Genetic Diseases

Dhruva Sahasrabudhe

(GENERAL AUDIENCE ABSTRACT)

Human beings suffer from a large number of rare genetic diseases. For many of these diseases, we do not know the genes which may play some role in the occurrence of a disease in a person, called gene associations for that disease. Computational methods to identify potential gene associations for diseases are useful. However, a majority of existing computational techniques to predict disease-gene associations require us to already know some of the genes which are associated with a disease, in order to find new potentially associated genes for that disease. This drawback becomes more serious for rare genetic diseases, since many of them do not have any known gene associations. In this work, we develop a computational method which can predict disease-gene associations even for diseases which have no already known gene associations, using information about similar diseases. We define the similarity of diseases using the observable biological characteristics of diseases. We also use information about the interactions between the proteins encoded by human genes, to make predictions about potential gene associations for a disease. Our method is designed using techniques from the field of graph machine learning. We show how our predictor performs, compared to other approaches, and analyze some of the choices made in its design, along with some properties of the outputs of our approach.

Dedication

Dedicated to my parents, and my grandparents. I am eternally grateful for the opportunities they have afforded me in life, through their hard work and sacrifice.

Acknowledgments

Firstly, I would like to thank Murali for helping me realize how science is meant to be done. Without him advising me, funding my studies and research through a GRA, helping me find a thesis topic, critiquing my writing and guiding me during weekly meetings, I would not have been able to do work which is (hopefully) worthy of a Master's thesis. I would also like to thank the members of the Computational Systems Biology Lab at Virginia Tech, for directly or indirectly teaching me how to read research papers, think critically, and present my thoughts and ideas with clarity and confidence. The group meetings and discussions with them have been a great source of knowledge for me. Most importantly, I would like to thank my family, and all my friends at Virginia Tech, in India, and in other parts of the world, without whom I would not be the person I am.

Contents

List of Figures	x
1 Introduction	1
1.1 Rare Diseases	1
1.1.1 Paradox of Rarity	2
1.1.2 Genetic Basis of Rare Diseases	2
1.1.3 Importance of finding Disease-Gene Associations for Rare Diseases .	3
1.2 Non-Computational Methods for Finding Disease-Gene Associations	4
1.3 Drawbacks of Non-Computational Methods for Rare Diseases	5
1.4 Computational Methods for Finding Disease-Gene Associations	6
1.5 Drawbacks of Computational Methods for Rare Diseases	6
1.6 Utility of Graph Machine Learning for Disease-gene Association Prediction .	7
1.7 Contributions of this Thesis	8
2 Review of Literature	10
2.1 Network Properties of Diseases and Genes	10
2.2 Using the Interactome for Disease-gene Association Prediction	12
2.3 Using the Human Phenotype Ontology as a data source	14

2.4	Integrative approaches for disease-gene prediction	15
2.5	Machine Learning Based Disease-Gene Prediction Algorithms	16
2.6	Graph Convolutional Networks	19
3	Data and Features	21
3.1	Data Sources	21
3.1.1	The Human Phenotype Ontology (HPO)	21
3.1.2	The STRING Interactome	23
3.2	Associations	23
3.2.1	Disease-Phenotype and Disease-Gene Associations	23
3.2.2	Gene-Gene Associations	24
3.3	Guiding Observations	24
3.3.1	Using Phenotypes for the Disease Similarity Measure	25
3.3.2	Using the Interactome for Novel Disease-Gene Associations	25
3.4	Features	27
3.4.1	Disease Features	27
3.4.2	Gene Features	29
4	Evaluation	31
4.1	Cross Validation	31
4.1.1	Sampling of Negatives	32

4.1.2	Cross-Validation Strategy	32
4.2	Ranking Based Evaluation	33
4.3	Disease Specific Evaluation	34
5	Algorithms	35
5.1	Link Prediction Approaches	35
5.1.1	Graph Convolutional Network	37
5.1.2	Common GCN Encoder	39
5.1.3	Multimodal GCN Encoder	41
5.1.4	Decoder	41
5.1.5	Loss function	44
5.2	Supervised Learning Approaches	44
5.2.1	Predictors	45
5.3	Cardigan	45
5.3.1	Modifications to Cardigan	47
6	Results	50
6.1	Per-Association Cross Validation	50
6.2	Per-Disease Cross Validation	52
6.3	Effects of Design Choices	54
6.3.1	Effect of Number of Nearest Neighbors in Disease-Similarity Graph	54

6.3.2	Effect of Different Gene Feature Vectors	55
6.3.3	Effect of Using the RESCAL Decoder	56
7	Analysis and Discussion	62
7.1	Embedding Analysis	62
7.2	Prediction Consistency Analysis	65
8	Conclusion and Future Work	71
8.1	Future Work	72
	Bibliography	75

List of Figures

3.1	HPO DAG structure	22
3.2	Jaccard Similarity Distribution of Disease Phenotypes	26
3.3	Shortest path length distribution of disease genes	27
5.1	Common GCN Encoder	40
5.2	Multimodal GCN encoder model	42
6.1	Per-Association Results	58
6.2	Per-Disease Results	59
6.3	Effect of Nearest Neighbors on AUPRC and early precision	60
6.4	Effect of Input Features on AUPRC and Early Precision	60
6.5	Effect of using a RESCAL Decoder on AUPRC and Early Precision	61
7.1	Cosine similarity distribution of final embeddings and input features	64
7.2	Cosine similarity distribution of final layer embeddings of positive and negative disease gene pairs	65
7.3	Inner product distribution of the Common and Multimodal GCN final embeddings	66
7.4	Heatmap for the disease classes of Cardigan, compared to Multimodal GCN	69

7.5	Row-normalized heatmap for the disease classes of Cardigan, compared to Multimodal GCN	70
-----	---	----

Chapter 1

Introduction

In this chapter, we introduce the problem this work tries to tackle: predicting genes associated with rare diseases. We give background information about rare diseases, which will enable the reader to understand why solving this problem is a legitimate and pressing public health concern, and more so, why computational methods may prove useful for this problem. We motivate the method we propose by describing properties both of rare diseases and of human genes that can aid us in solving this problem. We also describe the limitations and shortcomings of existing computational predictive methods that our method aims to overcome. We conclude by stating the research questions this work aims to solve, and by giving a formal problem definition.

1.1 Rare Diseases

Rather than being defined in terms of some biological criterion, a rare disease is categorized as such, by the frequency of its occurrence within the population under consideration. In the United States of America, a rare disease is defined as a condition which affects fewer than 200,000 people. According to population data obtained from the US census website [\[3\]](#) on the 19th of May, 2020, this means that a rare disease is defined as being a disease which affects no more than 1 person in roughly 1,648 people. According to the European Union, a rare disease is defined as a disease which affects no more than 5 people in every 10,000 people

(or 1 in every 2,000 people) [54]. In Japan, the threshold is 3.9 people in every 10,000 [41].

1.1.1 Paradox of Rarity

Rare human diseases pose a unique public health problem. The estimates for the total number of rare diseases in humans vary wildly, ranging between 5,000 and 8,000 [23, 41]. There is a large variation in the number of rare diseases for two reasons. Firstly, there are different definitions of what constitutes a rare disease, as we have seen above. Secondly, some sources do not count rare diseases arising due to chromosomal disorders, structural variations, and environmental effects. Moreover, different sources have different terminologies for the same disease entity [23]. However, recently there have been efforts to make consistent and complete knowledge-bases for rare diseases and their associated information [30].

Despite each rare disease by itself being rare, the number of people suffering from **some** rare disease is quite high. This phenomenon is termed as the *paradox of rarity* [39]. In fact, between five and ten percent of the human population suffers from some disease which can be termed as a rare disease [23] according to the definitions outlined above. It is estimated that more than 25 million Americans suffer from some rare disease, or 1 in 10 Americans [73]. This number is higher than the total number of Americans afflicted with AIDS and cancer combined [25, 55]. Using the European definition of a rare disease and conservative estimates of prevalence of rare diseases globally, it is estimated that between 263 and 446 million persons globally are afflicted by some rare disease [41, 89].

1.1.2 Genetic Basis of Rare Diseases

Although there do exist infectious or environmental rare diseases, almost **80%** of rare diseases are *genetic* [41]. A *genetic disease* is a disease whose manifestation can be attributed to one

or more abnormalities in the DNA of an individual. Gene variants whose presence in the genome of an individual either predisposes them to a disease, or which are directly responsible for the development of a disease are known as *disease-associated genes* [64].

It should be noted that this is a broader category than *disease-causing genes*, since disease-associated genes may not necessarily be causal for a disease. In the case of rare diseases, disease-associated genes may be rare gene variants present in the human gene pool and inherited from parent to offspring, or may be caused by rare mutations in certain specific genes [20, 67].

1.1.3 Importance of finding Disease-Gene Associations for Rare Diseases

Ascertaining the gene associations of diseases can help in prenatal or postnatal diagnosis [67]. Especially in the case of rare diseases, where cases are often misdiagnosed, or go undiagnosed [85], knowing the genes associated with the rare disease would greatly improve diagnosis. Knowing disease associated genes would also help us understand the mechanisms through which a disease manifests itself. This may open up avenues for treatment of the disease [53]. This can be through molecules that counteract the effects of the disease, or through externally introduced proteins which can compensate the function of the protein encoded by the missing or malfunctioning disease causing gene [61]. Less commonly, replacements of the disease causing gene, or genes which inactivate the disease causing gene, or which fight the disease symptoms can directly be supplied into cells externally in the form of RNA or DNA, to be translated into proteins by the cell itself, in an approach called *gene therapy* [81, 88].

Pharmaceutical companies are deincenivized from developing drugs for rare diseases, since

the market for each drug is small, and the number of rare diseases is high. Despite the Orphan Drug Act passed in the US in 1983 which provides economic incentives and tax breaks to drug companies for the development of drugs for rare diseases, up to 95% of rare diseases do not have FDA approved drugs [26]. Knowing disease associated genes helps identify the protein interaction pathways which enable the manifestation of the disease characteristics. The proteins along these pathways can be targeted by drugs [53].

1.2 Non-Computational Methods for Finding Disease- Gene Associations

Linkage analysis and *Genome-Wide Association Studies* are commonly used to identify disease associated genes. They rely upon the fact that genes which are in a similar location on the chromosome are often inherited together during meiosis. Genome-wide markers, which are segments of DNA that are known to occur at specific regions of the chromosome for a large fraction of the human population, are used in this method. The presence or absence of markers in the genome of disease-afflicted people can be compared to the markers in the genome of people known not to suffer from the disease. The genes in the areas of the genome around the markers found in disease sufferers, but absent from people not afflicted with the disease, are identified as potentially disease associated genes. [8, 69, 74].

These techniques are invaluable for narrowing down and identifying the regions of the genome associated with a large number of diseases, and have been particularly useful in finding the causal gene for many common monogenic Mendelian diseases [74]. They will continue to be powerful tools for identifying disease genes in the future, especially with the falling cost of whole genome and exome sequencing [76].

1.3 Drawbacks of Non-Computational Methods for Rare Diseases

Despite many advantages, linkage analysis or GWAS suffer some drawbacks. A GWAS or linkage study might identify huge regions of the genome as being disease-associated, while most of the genes in these regions may not be causative for or associated with the disease at all [15, 49]. GWAS or linkage studies are less effective for complex or multifactorial diseases [14]. Since many (individually rare) gene variants can cause the same set of disease characteristics, the genome regions identified by these means may not be complete, since they depend on the genomes which have been sequenced [68].

Sequencing a large number of genomes is expensive, and tracking down disease sufferers is also prohibitive in terms of time, effort and cost [76]. This is especially true in the case of rare diseases, since there may not be many sufferers in a particular area or country. Cross country collaboration may be needed to obtain enough individuals for linkage analysis in the case of rare diseases. Moreover, some rare diseases are multifactorial, and have a complicated pattern of inheritance, so linkage studies may not be useful for these diseases [15, 68]. Finally, the sheer number of rare diseases makes it challenging and expensive to use linkage studies and genome wide association studies to find the causal genes responsible for every rare disease. Some diseases may receive more attention and funding than others, leading to even more disparity in knowledge about disease associated genes.

1.4 Computational Methods for Finding Disease-Gene Associations

Computational methods to associate genes with diseases are a useful and powerful way to augment the GWAS/Linkage Analysis techniques discussed above [57, 67]. Computational disease-gene prediction methods are cost-effective, and scalable to many diseases, in stark contrast to GWAS or linkage analysis, which needs to be conducted separately for each disease. Moreover, computational predictive methods can integrate multiple data sources to prune the set of potential gene candidates for a disease, or find candidates which GWAS or linkage studies may miss, making them a valuable tool for the disease-gene discovery process [63]. We shall avoid discussing existing computational methods for predicting disease-gene associations in this section, since we cover them in detail, in Chapter 2.

1.5 Drawbacks of Computational Methods for Rare Diseases

A major drawback of a number of existing computational methods to predict gene associations for a disease is that they require some known gene associations for that disease as a starting point, in order to find new associations. Novel associations are found based on genes which are similar to, or interacting with known disease genes [7, 67, 91].

Since many rare diseases are understudied, they have no known disease genes. Many of these approaches fail when it comes to predicting genes for rare diseases. In fact, an analysis of the disease-gene associations in the rare disease database, ORPHANET [2], shows that over 47% of rare diseases do not have **any** disease genes associated with them. This is in stark

contrast to the set of common Mendelian diseases, in which almost **every** disease has at least some candidate causal genes.

Thus, methods which can prioritize genes for diseases with no known genes associations would be useful for predicting rare disease-gene associations. The majority of currently existing approaches which are able to make such predictions integrate data from multiple heterogeneous sources, in order to “transfer” associations from similar diseases with known associated genes.

1.6 Utility of Graph Machine Learning for Disease-gene Association Prediction

Recent years have seen an explosion in the number of graph machine learning based methods [93]. While traditional machine learning approaches assume that the input data are independent and identically distributed (i.i.d), graph machine learning based approaches aim to leverage relationships between data points, to propagate feature information among them [102]. This is based on the concept of *homophily* [51] in graphs, the idea that similar nodes tend to be connected by edges in many real world networks, and thus should have similar features.

Graph machine learning techniques have been developed for all sorts of supervised, semi-supervised and unsupervised predictive tasks over graphs, including node classification, graph classification [98, 101], link prediction [32, 78, 103], and graph embedding/clustering [12, 90].

Graph convolutional techniques extend the idea of convolutional neural networks (CNNs) [40] to graph structured data. While traditionally, the idea of convolution is applied to 2-D reg-

ular grid structured data like images (which can be thought of as 2-D grids of pixels), graph convolutional networks extend this idea for graph structured data, where the connections are irregular. The very broad goal of graph convolutional networks is to aggregate the features of a node with those of its neighbors [93].

In bioinformatics, graph machine learning, and especially graph convolutional techniques have been used for a number of predictive tasks ranging from protein function prediction [99], MicroRNA-drug association prediction [28], protein interaction prediction [83], protein interface prediction [17] cell growth simulation [48] and gene regulatory network reconstruction [72]. For a review of some techniques which use graph machine learning for tasks in bioinformatics, the reader can refer to Li *et al.* [43].

While there do exist graph convolution based techniques for disease-gene association prediction, their potential for this task has not yet been fully explored. We explicitly discuss graph machine learning/graph convolution based methods for disease-gene association prediction, and other graph convolution based methods which are directly relevant to this thesis in Section 2.6.

1.7 Contributions of this Thesis

In this chapter, we have defined rare diseases, disease genes, and the importance of finding the genes associated with rare diseases, in Section 1.1. We have described existing non-computational methods for discovering disease genes in Section 1.2, and discussed their drawbacks in Section 1.3. We have described computational methods for disease-gene association prediction in Section 1.4, and highlighted a major drawback of many computational approaches (their inability to make predictions for diseases with no known associated genes) in Section 1.5. We discussed how this drawback is even more apparent in the case of rare

genetic diseases. To the best of our knowledge, among the methods we have discussed in Chapter 2, only Cardigan, developed by Cáceres *et al.* [10] has been designed keeping this drawback in mind. It explicitly demonstrates results on disease-gene prediction for a set of diseases where no gene associations of the diseases to be evaluated are known by the algorithm.

In Section 1.6 we have described the growing field of graph machine learning, which has shown promise for predictive tasks in network biology. We discussed the potential of graph convolutional networks (a sub-domain of graph machine learning) for disease-gene association prediction. These observations culminate in the following research question:

Primary Research Question: Can we develop an approach based on graph convolutional networks, which can predict gene associations for a disease, even when there is no prior information provided to the algorithm about the genes associated with that disease?

Our aim is to see whether we can design a graph convolution based approach which outperforms other algorithms which address the same research question, given the same inputs, using a common evaluation scheme.

Chapter 2

Review of Literature

In this chapter, we provide an overview of the literature on computational prediction of disease-gene associations. We first describe findings that give us insight into the biological properties of diseases and genes in Section 2.1, which we can leverage to predict novel disease-gene associations. We also highlight some methods which make use of the same types of data that we use in this work, namely, methods which use protein interaction networks in Section 2.2, and methods which use the Human Phenotype Ontology (HPO) [34] in Section 2.3 (See Chapter 3 for information on our data sources.)

The method that we aim to develop (See Section 5.1) is a graph machine learning based method which integrates more than one data source. For these reasons, we discuss methods which integrate multiple data sources in Section 2.4, methods which use machine learning for disease-gene association prediction in Section 2.5, and then more specifically, methods belonging to the burgeoning field of graph machine learning in Section 2.6.

2.1 Network Properties of Diseases and Genes

An interactome, which is a network depicting a global view of all protein interactions in an organism, is a vital tool in computational biology. The nodes in the network correspond to proteins. The edges represent interactions between pairs of proteins. Protein interactions may be *physical* or *functional* [82]. Physical protein interactions represent protein complex

formation, or protein modifications through transfer of molecules. Functional interactions may be indirect, but represent that the two proteins are involved in the same steps of a biological process.

In the network paradigm, a genetic disease can be understood as a “perturbation” in the interactome. The expression of the protein encoded by a disease gene, and the interactions the expressed protein has with other proteins inside the body gives rise to, or “triggers” the phenotypes of a disease [6]. Using biological networks to gain insights into human diseases or to develop drugs is known as *network medicine*. Barabási *et al.* [6] review a few key hypotheses of network medicine, including: 1) the local hypothesis, which states that proteins involved in the same disease are likelier to interact with each other, 2) its corollary, which states that mutations in interacting protein lead to similar disease phenotypes, and 3) the disease module hypothesis, which states that the proteins associated with a disease form a *module*, a tightly connected functional or physical cluster. These hypotheses are explored in a number of other works, as we see below. For more information, the reader can refer to Chan *et al.* [13], who provide a detailed study into the role of network medicine in understanding human diseases.

Many observations have been made about the network properties of diseases and genes. While some earlier studies [18, 95] show that disease genes have a higher degree on average than random genes, Goh *et al.* [21] show that disease genes are actually less likely to be interactome “hubs” (i.e., high degree nodes). Feldman *et al.* [16] show that most disease genes have an intermediate degree. For most diseases, disease genes are non-hub nodes in the interactome. However, for a small fraction of diseases, especially cancers, they are hub nodes, with a high degree. Disease-genes are likelier to encode non-essential proteins (i.e. not critical to an organism’s survival), for most diseases. Disease genes are likelier to be expressed in only a few tissues, and genes associated with the same disease are expressed in

the same tissues [6, 29].

Goh *et al.* [21] construct a disease network by connecting diseases which share implicated genes, and show that while most diseases have a low degree, i.e., are connected to only a few diseases, some diseases like colon cancer have a very high degree. Jiang *et al.* [29] show that disease similarity networks (based on similarity of phenotypes) show high levels of clustering, and that the individual disease clusters correspond to disease categories.

Brunner *et al.* [9] show that diseases with similar phenotypes are associated with interacting genes or common genes. A number of works show that genes which are either physically or functionally interacting tend to have similar functions and similar effects on the manifestation of phenotypes, both in disease and in health [6, 7, 29, 45, 65]. However, Oti *et al.* [65] note that especially in the case of genes which perform multiple functions, different mutations can yield the manifestation of vastly different disease phenotypes, depending on which function of the gene is impaired.

2.2 Using the Interactome for Disease-gene Association Prediction

Network based methods that analyze the interactome to discover novel disease-gene associations range from simple techniques, such as using the direct neighbors of known disease-associated genes or shortest paths from such genes to more complex methods like diffusion kernels or random walks with restarts.

Oti *et al.* [66] use the direct neighbors of known disease genes in an interactome, along with information about genomic loci for diseases, to find disease gene candidates. Krauthammer *et al.* [35] use a shortest path based algorithm for disease-gene association prediction, assigning

higher scores to genes which are closer to a known “seed” disease-associated gene.

Hristov *et al.* [27] develop a guided random walk based approach for disease-gene discovery. For a complex or multifactorial disease, they incorporate “old” information about genes which are well-established as being linked with a disease, with “new” information about potential disease-associated genes. They perform a random walk with restart, beginning at the “new” genes, where the transition probability of the walker is weighted prior to the random walk, using a diffusion process from the “old” genes. This incentivizes the walker to visit nodes closer to “old” genes. In this way they combine both network diffusion and random walk based methods. They apply their method to predict disease genes for different types of cancer as well as three rare multifactorial diseases.

Navlakha *et al.* [59] compare various network based algorithms for disease-gene prediction, including direct neighbours, unsupervised clustering, semi-supervised clustering, random walks, and flow propagation. They find that each method is capable of making novel true predictions which the other methods miss, and taking a consensus of model outputs performs better than each individual method. This suggests that ensemble based techniques might be useful for this task. They show that the random walk with restart individually performs the best. Moreover, they observe better results for diseases whose genes display homophily (i.e. they are functionally similar or physically interacting), and state that for diseases which do not follow this rule, integrative approaches may be useful.

2.3 Using the Human Phenotype Ontology as a data source

A host of disease-gene prediction approaches make use of the Human Phenotype Ontology (HPO) in particular, as a structured, consistent data source mapping diseases to phenotypes. Many of these methods use this disease-phenotype mapping, along with the ontological structure of the HPO to obtain a semantic similarity measure between pairs of diseases.

Masino *et al.* [50] prioritize genes for the clinical phenotype of a patient by comparing the HPO terms describing the patient phenotypes with the HPO terms associated with genes. Oellrich *et al.* [62] define a phenotype similarity mapping between the HPO and the Mammalian Phenotype Ontology (MPO) [80]. They use disease-gene associations studied in mouse models to prioritize genes for phenotypically similar diseases in humans.

Le *et al.* [37] perform a comparative study of using different ontologies for disease-gene association prediction, disease-miRNA association prediction, and disease module prediction. As part of their analysis, they compare the effects of using an HPO based disease similarity measure [75], as opposed to using the so-called Van Driel similarity matrix based on text mining of the OMIM [84]. They construct a five nearest-neighbor disease network by integrating each disease similarity measure. In their work, they find that the HPO based matrix gives demonstrably better results for the prediction tasks. Le *et al.* [38] show that an HPO based disease similarity measure similarly outperforms a Disease Ontology based measure for the task of predicting disease-lncRNA (long non-coding RNA) associations.

In contrast to methods described above, Shen *et al.* [77] use associations between HPO terms and diseases to predict relationships between HPO terms. They obtain HPO term-disease associations from OMIM, Orphanet, and Decipher to derive enriched HPO term embeddings.

They add edges to the HPO by connecting terms associated with the same disease, and use the Node2vec algorithm [22] to learn node embeddings for HPO terms through a random walk over the enriched graph. For evaluation, they use the learned embeddings to predict (positive or negative) associations between pairs of HPO terms.

2.4 Integrative approaches for disease-gene prediction

Many approaches integrate genomic data, such as GO annotations, and protein sequences, on the assumption that disease-genes would have similarities in these properties, which could be used to find new disease-gene associations.

Aerts *et al.* develop the algorithm Endeavour [4], which integrates data from 10 different sources to construct gene features. These sources included gene expression data, references in the literature, function, sequence similarity, etc. They use the constructed features to rank candidate genes for a disease based on similarity to genes already associated with the disease.

Wu *et al.* develop the algorithm CIPHER [92] that constructs an integrated network of diseases and genes. To score a query disease-gene pair, the algorithm uses both the phenotypic similarity of diseases to obtain a similarity profile for the disease and the interactome to obtain a gene closeness profile for the gene. By feeding these into a linear model, it obtains a final score for a disease-gene pair.

Li *et al.* [42] extend the random walk with restart algorithm for a heterogeneous network of diseases and genes, for simultaneous disease and gene prioritization, connecting diseases using a 5 nearest-neighbor disease-disease network from the Van Driel disease-similarity matrix.

Vanunu *et al.* develop the algorithm PRINCE [86], which finds unknown disease-gene asso-

ciations using known disease-gene associations, and the disease similarity matrix from Van Driel *et al.* [84]. Given a query disease, the score assigned to a gene in the interactome is determined by the similarity between the diseases it is associated with, and the query disease. A network propagation formula is used over the interactome to make scores smooth across the network, and assign scores to genes with no disease associations. Cardigan [10] is a more recent approach designed on similar principles, using the Caniza disease similarity matrix [11] instead of the Van Driel matrix, which demonstrates superior performance to PRINCE.

Mordelet *et al.* develop the semi-supervised algorithm ProDiGe [56]. ProDiGe uses the paradigm of PU (Positive-Unlabeled) learning, which involves learning from both positive and unlabelled examples, using the distribution of unlabelled examples to improve the predictor. It trains a biased SVM over disease-gene pairs by randomly sampling some unlabelled genes as negative genes for a disease, and also by randomly adding some unlabelled genes as positives for a disease to the set of known positive disease-gene pairs. A variant of ProDiGe can make predictions for diseases with no known associated genes, by using both a disease similarity kernel and a gene similarity kernel. In the evaluation conducted by the authors, ProDiGe is shown to outperform both PRINCE and Endeavour. Yang *et al.* [97] use an ensemble of PU learning algorithms to achieve superior results to ProDiGe.

2.5 Machine Learning Based Disease-Gene Prediction Algorithms

In this section, we give a brief history of machine learning based methods in disease-gene association prediction, going in a roughly chronological order.

Early machine learning methods in disease-gene association prediction formulate the problem as two-class binary classification, with positive and negative training examples. The positive examples are known disease-gene associations, while the negative examples are generally disease gene pairs (where the genes are sampled from the genes not known to be associated with a disease). However, the set of genes not associated with a disease also contains the disease-gene associations to be discovered, which raises issues regarding what the negative examples should be. A limitation of classification based disease-gene prediction algorithms is that the set of negative examples often has to be somewhat artificially constructed.

Some early disease-gene prediction algorithms aimed to identify whether a gene can be potentially disease causing, without explicitly mapping it to a particular disease. For example, López-Bigas *et al.* [46] use protein sequence information to train a decision tree algorithm to determine whether a gene may be disease-causing. Xu *et al.* [96] derive topological features for proteins from protein interaction networks, such as the degree and proximity to known disease-genes. They use a k -Nearest Neighbor classification method to find genes which may be disease-causing. Smalter *et al.* [79] derive features from topological information about the interactome, along with protein sequence information, to train a binary SVM which recognizes disease-associated genes associated with some disease.

As the field progressed, methods started aiming at predicting gene associations for specific genes. Radijovac *et al.* [70] derive features from the Disease Ontology, Gene Ontology, interactome, along with physiochemical/sequence information about proteins to train a separate binary SVM for each disease, in order to predict genes associated with it.

Next, some studies approached the problem using matrix factorization techniques, inspired by approaches used in recommendation systems. For example, Natarajan *et al.* [58] formulate the problem as matrix completion of a low rank disease-gene association matrix. They develop an inductive matrix factorization algorithm using disease and gene features from

diverse sources. It can make predictions for diseases which are not seen at training time in addition to making gene predictions for diseases with no known gene associations. Zeng *et al.* [100] present a probabilistic collaborative filtering model for disease-gene prediction.

Concurrently with matrix factorization based methods, semi-supervised learning methods were also found to produce competitive results for this problem. We avoid discussing these algorithms in this section, since we have covered many popular semi-supervised methods above in Section 2.4.

Recently, a number of methods have used deep neural networks for disease-gene prediction. Luo *et al.* [47] use a multimodal deep belief network (DBN) for the task. Their model contains two DBN submodels. The first submodel takes as input a feature vector for a disease-gene pair created using an interactome, while the second submodel takes as input a disease-gene pair feature vector created using the Gene Ontology (GO). The submodels transform the data belonging to two different feature space into a common latent representation. The latent features are then fed to a final submodel, which is used to predict the probability of a disease-gene association. Alshahrani *et al.* [5] learn disease and gene features in an unsupervised way, by using a skip-gram model [52] over the phenotypes associated with genes and diseases. They feed the learned features to a neural network to predict disease-gene associations.

The review by Le *et al.* [36] gives an overview of machine learning methods in disease-gene prediction. It details a roadmap of the development of disease-gene prediction algorithms, highlighting the transition of methods from binary classifiers, to one class methods, semi-supervised learning methods, deep learning, matrix factorization, and ensemble learning methods.

2.6 Graph Convolutional Networks

Rao *et al.* [71] develop an approach based on Graph Convolutional Networks (GCNs) [32] for rare disease-gene prediction. They create a heterogeneous network of phenotypes (obtained from the HPO), genes, diseases (from OMIM), and protein interaction pathways (obtained from Orphanet). The inter-phenotype edges are weighted using a phenotype similarity measure, while the disease-phenotype edges are weighted using the Orphanet disease-phenotype association score. All other edge weights are set to 1. Their approach, called GCAS, modifies the GCN algorithm to allow for effective information propagation over a larger neighborhood in the network.

Singh *et al.* [78] extend the Variational Graph Autoencoder proposed by Kipf *et al.* [33] in order to apply it for link prediction over heterogeneous networks. They evaluate their method for disease-gene association prediction. The algorithm involves constraining the loss function to predict associations between disease nodes and gene nodes in a heterogeneous network.

Zitnik *et al.* [103] propose DECAGON, which uses graph convolutional networks to predict the side effects of drug-drug associations, formulated as a link prediction problem over a multimodal graph of drugs and proteins. Protein-protein edges represent functional or physical interactions, obtained by combining multiple PPI networks. Drug-protein edges represent chemical interactions between the drug and the protein. Drug-drug edges are side-effect specific, representing that a pair of drugs have been observed to cause that particular side effect, obtained from adverse medical reporting data. They train separate graph convolutional encoders to aggregate information over the neighborhood of a node, under each edge type, where drug-protein edges, protein-protein edges, and each drug-drug side effect edge constitute distinct edge types. They train a set of tensor factorization decoders to predict

the probability of an edge of each type between any two nodes in the network. The model is trained end-to-end (i.e., the encoder and decoder are jointly optimized). Li *et al.* [44] adapt the same approach for disease-gene prediction. Their multimodal network consists of a disease similarity network using MimMiner [84], OMIM disease-gene mappings, and the HumanNet PPI network. Similar to Decagon, they train separate encoders for neighborhood aggregation over each edge type. However, they employ a single bilinear edge decoder, and compute loss only over the disease-gene edges, as opposed to Decagon, which computes loss over all edge types. Han *et al.* [24] combine graph convolution with matrix factorization for disease-gene prediction. Based on domain information about diseases and genes, they construct a reduced dimension feature vector, using which they construct a K-nearest neighbor disease similarity and gene similarity graph. They perform graph convolutions on this graph, to obtain latent representations of diseases and genes, which they decode using matrix factorization.

Chapter 3

Data and Features

In this chapter, we describe the sources and formats of the data that we use. We perform two simple tests to see whether certain properties of disease-phenotype associations and protein interaction networks which may be informative for the task of disease-gene association prediction, also hold true with our data sources. We aim to explore properties documented by previous works, discussed in Section 2.1. We go on to describe how we preprocess the raw data and convert them to disease and gene features for use in our predictors.

3.1 Data Sources

We consider two main sources of data in our work, the Human Phenotype Ontology (HPO) [34] and the STRING interactome [82].

3.1.1 The Human Phenotype Ontology (HPO)

A phenotype is an observable characteristic or behavior associated with a disease. The HPO provides a consistent, hierarchical vocabulary to describe human disease phenotypes and the relations among them. The HPO takes the form of a directed acyclic graph (DAG), encoding the relationships among roughly 13,000 human disease phenotypes. As an illustrative example, the relationship between six terms in the HPO is given in Figure 3.1. An

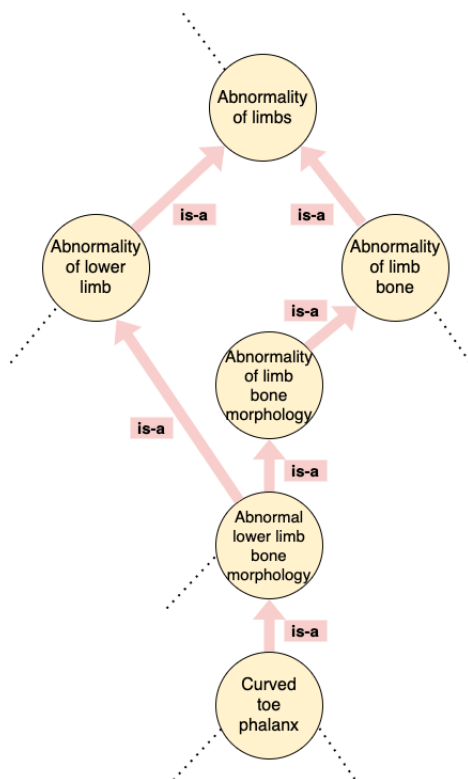


Figure 3.1: The DAG connecting six HPO terms.

edge in the HPO represents an *is-a* relationship between a more general parent phenotype, and its more specific child phenotype. For example, the term *Curved toe phalanx* **is-a** *Abnormal lower limb bone morphology*. Terms in the HPO can be annotated to proteins (or equivalently, the genes that encode these proteins), with a many-to-many relationship. This mapping is obtained from the biological literature using a variety of means, either through manual curation, or automatic electronic parsing of the publication record for the disease or phenotype [34]. We use the Human Phenotype Ontology (HPO) for disease-phenotype and disease-gene associations.

3.1.2 The STRING Interactome

We obtain the human protein-protein interaction network from the “Search Tool for Retrieval of Interacting Genes/Proteins” (STRING) database.

The edges in the human STRING network correspond to functional associations between proteins. These associations may be direct, like a chemical interaction between two proteins, or protein complex formation, or indirect, like a transcription factor regulating the expression of a protein. The human STRING network also contains edges between two proteins if they are involved in the same specific biological process. The STRING database integrates *primary* (through biological protein interaction experiments) and *predicted* (using computational tools) protein interactions. The sources of these interactions include seven channels, including text mining of literature, biological experiments, coexpression of proteins, data obtained from other protein interaction databases, and predictions using genomic context.

3.2 Associations

We use three types of associations in this work, namely disease-phenotype associations, disease-gene associations, and gene-gene associations.

3.2.1 Disease-Phenotype and Disease-Gene Associations

We obtain these data from the Human Phenotype Ontology (HPO) [34]. Both disease-phenotype associations and disease-gene associations have been obtained on the 19th of January, 2020. The HPO disease-gene associations are in turn obtained from two databases of disease-gene associations, OMIM [1] and Orphanet [2]. We filter out all the diseases which either do not have any gene associations, or whose gene associations are not present in the

STRING network, or are not associated with any HPO phenotype. This step results in 6,828 diseases with 9,539 disease-gene associations. We obtain a set of 198,984 disease phenotype associations.

3.2.2 Gene-Gene Associations

We obtain gene-gene associations from the 2019 Human STRING interactome network (v11) [82]. Since the genes in the HPO-gene associations are in the Entrez-Gene namespace, we map all the genes from the default STRING namespace to Entrez-Gene. The raw STRING network has weighted edges, with edge-weights (ranging from 150 to 1000) corresponding to the confidence of the interaction represented by the edge. We only keep the edges with score above 600, corresponding to roughly the 90th percentile of edge weight. This threshold keeps relatively high confidence edges in the network, without making the network too sparse. After preprocessing, we treat the network as an undirected, unweighted graph. The resulting network has 15,870 nodes and 899,411 unweighted, undirected edges.

3.3 Guiding Observations

Here, we explore some properties of disease-phenotype associations, in diseases which share genes, and some properties of genes which are associated with the same disease. Specifically, we test whether disease pairs which share gene associations are relatively more similar in terms of their set of phenotypes, compared to random disease pairs. We also test whether gene pairs associated with the same disease are closer in the interactome compared to random gene pairs.

3.3.1 Using Phenotypes for the Disease Similarity Measure

Our approach uses the assumption that diseases which share phenotypes are also associated with the same genes. Despite finding supporting evidence in our survey of the literature (see Section 2.1), we also perform some simple tests to ensure this assumption holds true on our data. To this end, we performed the following simple test: We identified 16,138 pairs of diseases which have at least one associated gene in common. For each such pair, we computed the Jaccard coefficient of their sets of associated phenotypes. We compared the distribution of Jaccard coefficients with that of an equal number of random pairs of diseases. Here, we assume that if a disease is associated with an HPO phenotype term, then it is also associated with all of the parent phenotypes (i.e. more general terms) in the HPO, since the disease-phenotype associations have a transitive relationship within the ontology. The results are shown in Figure 3.2. We see that the Jaccard similarity in the case of disease pairs with shared gene associations is **3.7 times** higher on average than that of random pairs of diseases. This supports the claim that phenotypically similar diseases are likelier to share gene associations. We propose to make use of this property to make predictions about gene associations for diseases with no known gene associations, using diseases that are phenotypically similar to it, and have associated genes.

3.3.2 Using the Interactome for Novel Disease-Gene Associations

We also explored whether genes associated with the same disease tend to be comparatively highly clustered in the STRING interactome. This claim is also well supported by the literature, as discussed in Section 2.1. To validate this property, we plot the distribution of shortest path lengths for all 22371 pairs of genes associated with the same disease, and compared them with the distribution of shortest path lengths for an equal number of random

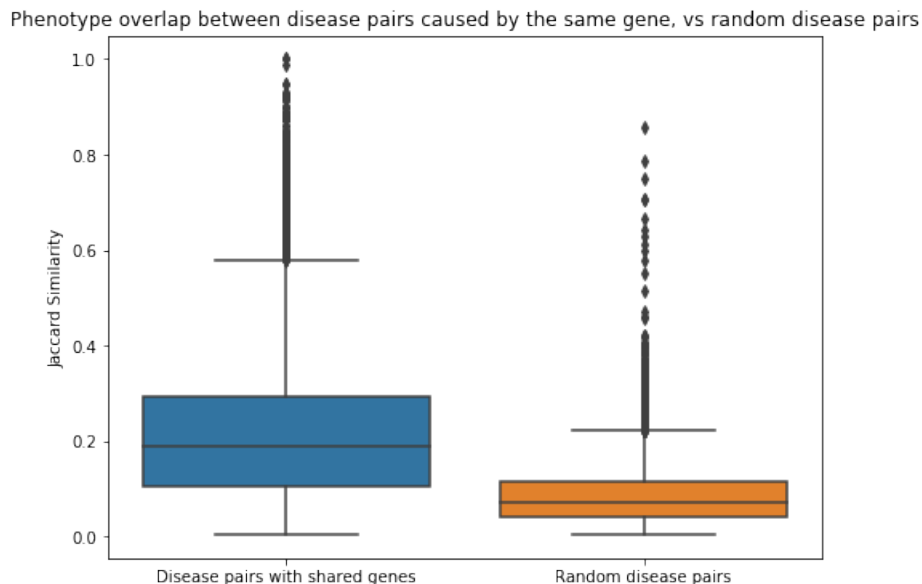


Figure 3.2: Distribution of Jaccard similarities of the set of HPO phenotypes of disease pairs with at least one shared associated genes compared to randomly-selected disease pairs.

gene pairs in the STRING interactome. The results are shown in Figure 3.3. We can see that 30% of gene pairs associated with the same disease are connected by a path of length one in the former case, in stark contrast to the latter, where the proportion is 100 times smaller, at 0.3%. While almost 89% of randomly chosen gene pairs are a path length of three or greater away from each other, more than 60% of the gene pairs associated with the same disease are either connected by an edge or by a path of length two. This analysis shows how invaluable the interactome is for locating additional disease-associated genes by exploring a small neighborhood (less than two hops) of genes which are known to be associated with a disease.

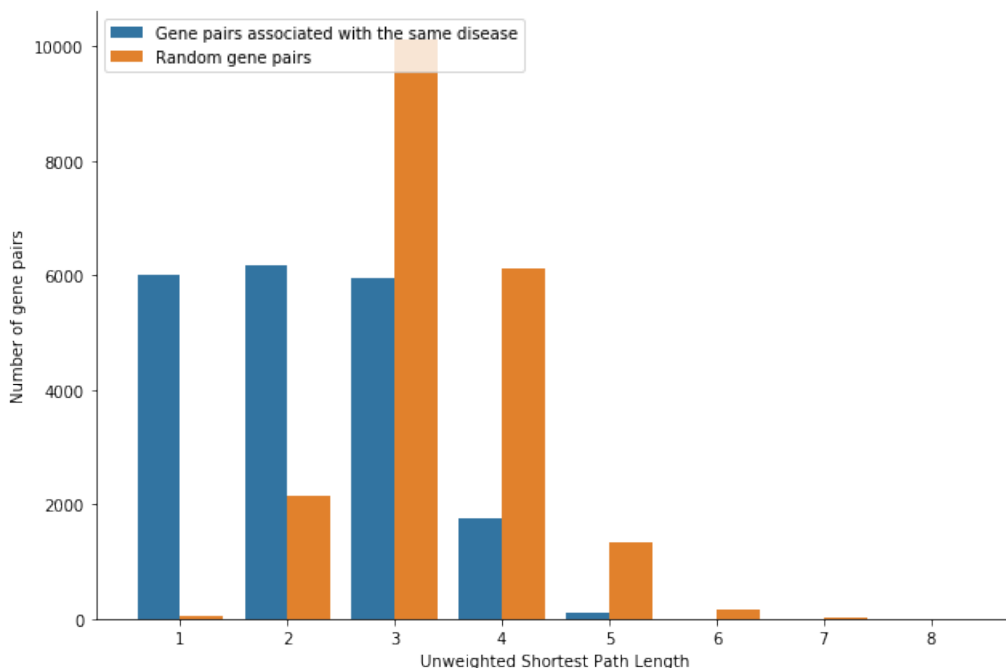


Figure 3.3: Distribution of shortest path lengths in the unweighted, undirected STRING interactome, between genes associated with the same disease, compared to random gene pairs.

3.4 Features

Now that we have described our data sources and given some evidence as to why these data sources may be useful to predict the genes associated with a disease, we describe the disease and gene features we use for the various predictors that we develop in Chapter 5.

3.4.1 Disease Features

The disease features are created using the disease-HPO term associations for each disease, using the data described in Section 3.1. We require some way to capture the similarity of the set of phenotypes of pairs of diseases, which takes into consideration the ontological structure of the HPO. We call this the *phenotypic similarity* of a pair of diseases. We use

the HPO Robinson similarity method [75], which is an information theoretic measure of disease similarity. It takes into account the set of phenotypes annotated to each disease, the structure of the HPO ontology itself, and the frequency of occurrence of the annotated disease phenotypes. We describe the Robinson disease similarity computation below.

We first need to formulate how to measure the similarity between two HPO terms. Given HPO terms t_a and t_b , with a set of ancestors in the HPO as T_a and T_b respectively. Let the fraction of disease terms annotated with HPO term x as $p(x)$. We define their similarity using the following equation:

$$\text{tsim}(t_a, t_b) = \max_{x \in T_a \cap T_b} \left(-\log(p(x)) \right). \quad (3.1)$$

tsim ranges between $[0, \infty)$, with similar terms having higher scores. Since all the disease annotations of a more specific term t_c are automatically assigned to its less-specific parent t_p , $p(t_c)$ is always less than $p(t_p)$ for all the HPO terms and their parent terms. Knowing this, we can see that Equation 3.1 gives high similarity values for pairs of terms which have common ancestors which are more specific (i.e. deeper terms in the ontology), rather than more general. Since the entire HPO has a common root term, it is guaranteed that each pair of terms will have at least one common ancestor, and thus will have some similarity value.

We can now define the similarity dsim of a disease d_a (with the set of HPO terms associated with it as T_a) and disease d_b (with the set of HPO terms associated with it as T_b) in the following way:

$$\text{dsim}(d_a \rightarrow d_b) = \text{avg}_{t_a \in T_a} \left(\max_{t_b \in T_b} (\text{tsim}(t_a, t_b)) \right) \quad (3.2)$$

Equation 3.2 is not symmetric. That is, $\text{dsim}(d_a \rightarrow d_b) \neq \text{dsim}(d_b \rightarrow d_a)$. We average the two, to obtain the final symmetric version of the disease similarity between d_a and d_b ,

$\text{dsim}_{\text{symm}}$:

$$\text{dsim}_{\text{symm}}(d_a, d_b) = \frac{\text{dsim}(d_a \rightarrow d_b) + \text{dsim}(d_b \rightarrow d_a)}{2}, \quad (3.3)$$

For our set of d diseases, we generate a $d \times d$ *Robinson similarity matrix* D . The entry in the a^{th} row and the b^{th} column of the matrix corresponds to the disease similarity $\text{dsim}_{\text{symm}}(d_a, d_b)$. The feature vector for a disease is of the dimension $d \times 1$, corresponding to the row associated with that disease in this matrix. We reduce the dimensions of this vector to 256 using Principal Component Analysis (PCA). We find that the raw disease feature vector is sparse and high-dimensional. Reducing the dimensions of the feature vector to 256 greatly reduces the number of parameters needed by the methods described in Sections 5.1 and 5.2, while preserving the variance of the data. About 97% of the variance of the disease features is captured by a 256 dimensional PCA of the disease similarity matrix). The final reduced dimension feature vector for each disease, denoted by ϕ_d captures its phenotypic similarity with every other disease in the dataset. We call this feature vector the **Robinson Similarity Vector** in this work.

3.4.2 Gene Features

For each gene g in the STRING network (described in Section 3.1), we define its feature vector to be its adjacency vector. Since we treat STRING as an unweighted graph, this vector is binary. For reasons similar to those given for disease feature creation in Section 3.4.1, we reduce the dimensions of this vector to 256 using PCA. The reduced dimension feature vector for gene g , denoted by ϕ_g is called the **STRING PPI Adjacency Vector** in this thesis. We also use a vector of all 0s for the gene features. Obviously, this would be useless for the supervised learning approaches described below in Section 5.2, since the model would not be able to discriminate between genes. However, for reasons we will delve into in the next

chapter, we use this feature vector for the methods described in Section 5.1. The feature vector for a gene g , denoted by ϕ_g^0 has the same dimensions as ϕ_d , and is called the **Blank Vector** in this thesis.

Chapter 4

Evaluation

In this chapter, we describe how we partition the disease-gene associations described in Section 3.2.1 for cross-validation. We also describe the evaluation measures that we use to compare the performance of the methods discussed in Chapter 5.

4.1 Cross Validation

We partition the disease-gene associations obtained from the HPO (See Section 3.1) into training (70%), test (20%), and validation (10%) sets. We create these partitions in two ways.

Per-Association Cross Validation. The first strategy involves partitioning disease-gene associations into uniformly at random into three sets, according to the desired proportion (70-20-10). Here, the associations of a particular disease associated with more than one gene may be broken up across multiple sets. Therefore, in this strategy, we mimic finding novel gene associations for a particular disease when some associated genes **may or may not** be known beforehand.

Per-Disease Cross Validation. The second strategy involves splitting the set of diseases themselves into a training, test and validation set according to the desired proportion (70-

20-10). Here, **all** the gene associations for a particular disease belong to only one of the three sets. This strategy corresponds to the case of finding novel gene associations for a particular disease when **no** associated genes are known beforehand. This way of partitioning the data is of more interest to us, since it corresponds to the fact that many rare diseases do not have any known associated genes (Chapter 1).

4.1.1 Sampling of Negatives

Given a positive integer r , for every positive (i.e., with label 1) disease-gene association (d, g^+) in the training, test or validation set, we randomly sample a set of r “negative” gene associations ν_d for d from the genes in the STRING interactome. For each gene $g^- \in \nu_d$ we add (d, g^-) to the corresponding dataset with label 0. Here, r is the desired ratio of negative to positive disease-gene associations. We present results for three values of r : 1, 5 and 10.

4.1.2 Cross-Validation Strategy

We use the training set to optimize the model parameters and the validation set to adjust hyperparameters (e.g., the number of layers and the number of parameters in the models, number of epochs after which training should be stopped, and the learning rate). We evaluate the algorithm performance on the test set.

We perform five iterations of *repeated random subsampling*, which involves partitioning the train, test, and validation sets five times with different random seeds. We report combined scores over all five iterations, by plotting a barplot of scores for the measures. Error bars on the barplots represent the 95% confidence interval. For each algorithm being evaluated, the positive and negative associations used in the training, test, and validation set are consistent across the five iterations.

After training, we evaluate the performance of the methods on the test set, using the evaluation techniques to be described below. The model takes a disease gene association as input, and outputs a score corresponding to the model’s confidence in whether the given disease-gene association is a true positive association.

4.2 Ranking Based Evaluation

The model output consists of scores for all the test disease-gene associations. These scores represent the model’s confidence in the disease-gene association being a true positive association, and can be used to rank the positive and negative associations across all disease-gene pairs in the test set in decreasing order of confidence. This global ranked list of disease-gene associations can be compared with the list of positives in the test set, to find the *precision* and *recall* at each rank, the *early precision* (EP), and the *the area under the precision recall curve* (AUPRC) for the entire ranked list.

These measures are calculated in the following way for the ranking task: Let there be n^+ positive and n^- negative disease-gene associations in the test set. The total number of test disease-gene associations are $n = n^+ + n^-$. Each method outputs a ranked list L of all n disease-gene associations in the test set, in decreasing order of model confidence. For each rank i in the ranked list L , let the number of true positive disease-gene associations between rank 1 and rank i in L be n_i^+ . Using this notation, we define:

- $\text{precision@}i = \frac{n_i^+}{i}$
- $\text{recall@}i = \frac{n_i^+}{n}$

We use these measures to define two indicators of model performance, namely the early precision (EP), and the area under the precision recall curve (AUPRC). The EP corresponds

to the value of precision at the number of true positives in the training set. Thus the early precision $EP = \text{precision}@n^+$. The area under the curve of $\text{recall}@i$ vs $\text{precision}@i$ across all $1 \leq i \leq n$ gives us the AUPRC.

4.3 Disease Specific Evaluation

Since the ranking task based evaluation considers all diseases together, we also define a disease-specific evaluation measure. We do this since it may be the case that for some diseases, the model score may be low for the true positive gene association, leading the true positive disease-gene to be ranked low in the ranking task. However, the model score for the negatively sampled genes for that disease may be **even lower**. Thus, in a ranking of genes for that disease, the true positive would still show up first. In this measure, we report the fraction of times the true positive disease-gene pair for a disease was ranked higher than any randomly sampled negative disease-gene pair for the same disease.

Chapter 5

Algorithms

In this chapter, we begin by describing our contributions, which are based on graph convolutional networks, in Section 5.1. We follow up by describing the supervised learning classifiers we implement as simple benchmarks for our final methods in Section 5.2. We conclude by describing *Cardigan* [10] in Section 5.3, a recent, state-of-the-art network propagation approach developed by Cáceres *et al.*. We use Cardigan as a competitive benchmark for the task of disease-gene association prediction. We describe the modifications we make to the algorithm in order to run it on the data described in Chapter 3 and according to the evaluation procedure outlined in Chapter 4.

5.1 Link Prediction Approaches

In this work, we formulate the association prediction task as a link-prediction problem over a multimodal network of diseases and genes. The multimodal network is a union of three networks; a *gene-gene network*, a *disease-gene network*, and a *disease-disease network*. The gene-gene network refers to the STRING interactome mentioned in Section 3.1. The disease-gene network is a bipartite graph, where nodes correspond to diseases and genes, and edges correspond to the disease-gene associations obtained from the HPO, described in Section 3.1. The disease-disease network is constructed using the Robinson Similarity Vector of each disease (see Section 3.4.1). For a pair of diseases, the Euclidean distance between the Robinson

Similarity Vectors of the pair is calculated. Each disease is connected by an unweighted, undirected edge to the five diseases whose feature vectors have the smallest Euclidean distance from its own feature vector. The rationale for constructing the disease-disease network is given below

Thus, are two types of nodes in this network, disease nodes and gene nodes. They are connected to each other by three types of edges, disease-disease edges, disease-gene edges, and gene-gene edges.

To perform link prediction, we develop what can be termed as **end-to-end trained graph convolutional encoder-decoder** models. A broad overview of this approach is described below:

- The input to the algorithms includes the multimodal network described above, with the disease nodes and gene nodes having some version of the initial disease/gene node features described in Section 3.4.
- We then perform a series of *graph convolutions*, which are a form of parameterized neighborhood aggregation, over the network, to obtain an *embedding* for every single node in the network. The embedding is a dense vector $\in \mathbb{R}^f$, where f is the output dimension of the final layer of convolution.
- For every pair of (disease, gene) nodes in the graph, we *decode* the embeddings of the pair of nodes to obtain a probability of an edge between the (disease, gene) pair.
- During training, we adjust the parameters of the encoder so as to maximize the probability of true positive disease-gene edges in the training set, while also minimizing the probability of edges between negatively sampled disease-gene node pairs.
- Once the model is trained, we process every disease-gene association in the test set

using the encoder and the decoder to obtain a score.

5.1.1 Graph Convolutional Network

We use a Graph Convolutional Network (GCN) [32] as the basis for our graph convolutional encoder. The GCN is a widely-used and computationally fast technique to perform parameterized neighborhood-based feature aggregation over the nodes in a network. It is based on the concept of the first order approximation of the graph Fourier transform. We describe a single layer of the GCN convolution below.

We are given a network $G = (V, E, W)$, consisting of a set of vertices V of size v , a set of edges E of size e and the $v \times v$ matrix of edge weights W , where if $(i, j) \in E$, where W_{ij} corresponds to the weight of the edge between node i and node j . Moreover, we are given a node feature matrix, ϕ of dimensions $v \times h_0$, where each column corresponds to a h_0 dimensional feature vector for a node in V . A single layer of GCN convolution transforms the input feature vector for each node in V from dimensions h_0 , to an *embedding* of dimension h_1 , where h_0 does not necessarily equal h_1 . Multiple layers of GCN convolutions can be stacked, each layer treating the embeddings generated by the previous layer as input, transforming them to a new embedding with a different dimensionality. Both the number of layers, and the dimensions of the output of each layer are hyperparameters of the method. Mathematically, a single layer of convolution, for some arbitrary GCN layer t is given below.

Let H^t refer to the $v \times h_t$ input embedding matrix (i.e., the GCN output in the layer $t - 1$). For $t = 0$, H^0 is simply the initial node feature matrix ϕ . The output of GCN layer t is H^{t+1} , a $v \times h_{t+1}$ dimensional embedding matrix.

Let $\hat{G} = (V, \hat{E}, \hat{W})$ be the result of adding self-loops of weight 1 to every node in graph G . That is, $\hat{W} = W + I$, where I is the $v \times v$ identity matrix. Let $\hat{\delta}$ be the degree matrix of $\hat{G}$. $\hat{\delta}$

is a $v \times v$ diagonal matrix, where $\hat{\delta}_{ii} = \sqrt{\sum_{k:(i,k) \in \hat{E}} \hat{W}_{ik}}$. A few choices of non-linear functions σ include the sigmoid function, where $\sigma(x) = \frac{1}{1+e^{-x}}$, or Rectified Linear Unit (ReLU), where $\sigma(x) = \max(0, x)$. When applied to a matrix, or vector, σ is applied separately to each entry. $\theta^{(t)}$ is a trainable parameter matrix, of dimensions $h_t \times h_{t+1}$, for layer t . The GCN equation is as follows:

$$H^{(t+1)} = \sigma(\hat{\delta}^{-1/2} \hat{W} \hat{\delta}^{-1/2} H^{(t)} \theta^{(t)}) \quad (5.1)$$

To understand the effects of this transformation to the embeddings of a single node, we look at the vectorized version of this equation. For a single node j with embedding h_k^t as input to layer t , the node embedding of j after the GCN convolutional layer, $h_j^{(t+1)}$ is given by:

$$h_j^{(t+1)} = \sigma \left(\sum_{k:(j,k) \in \hat{E}} \frac{1}{\sqrt{\hat{\delta}_{ii} \hat{\delta}_{jj}}} h_k^{(t)} \theta^{(t)} \right) \quad (5.2)$$

We see that for each node, a single convolutional layer is tantamount to an aggregation, parameterized by $\theta^{(t)}$ of the input features of its direct neighbors (including itself, due to the addition of self loops), scaled by a constant factor $\frac{1}{\sqrt{\hat{\delta}_{ii} \hat{\delta}_{jj}}}$, in accordance with the degree of both the node and its neighbor. The aggregated vector is then transformed by a non-linear function σ . Thus, a single convolutional layer combines node feature information over a first order neighborhood of a network. Chaining together multiple convolutional layers allows us to aggregate information from distant nodes, more hops away in the graph.

5.1.2 Common GCN Encoder

In this approach, we treat every edge type (disease-disease, disease-gene, and gene-gene) as the same. We train a common GCN encoder for neighborhood aggregation of the features over every node and every edge in the graph, be it a disease or gene. Information propagation occurs between disease pairs through disease-disease edges, gene pairs through gene-gene edges, and between diseases and genes through disease-gene edges.

Our architecture consists of 3 layers of convolution, with a 64-dimensional GCN in each layer.

A drawback of this approach is that if the initial features of diseases and genes are of a different number of dimensions, or represent different feature spaces, a GCN which treats all edge types and node types the same cannot be used. Thus, in this work, since the disease features are 256-dimensional, we choose to use a 256-dimensional Blank Vector over all gene nodes as their initial feature vector. While this would not be a sensible choice for gene features in the case of the supervised learning approaches described in Section 5.2, it is reasonable in the graph convolutional case. In the first layer of convolution, feature information “flows” to the genes in the network, from the diseases each gene is associated with, through the disease-gene edges in the multimodal network. After the first layer of convolution, the gene features represent a parameterized aggregation of the feature vectors of its associated diseases. This information is then propagated to other genes in the network through successive rounds of convolution. This strategy is an effective way to transmit feature information from diseases to the genes they are associated with, and then successively to genes in their neighborhood. A graphical representation of the Common GCN Encoder is shown in Figure 5.1.

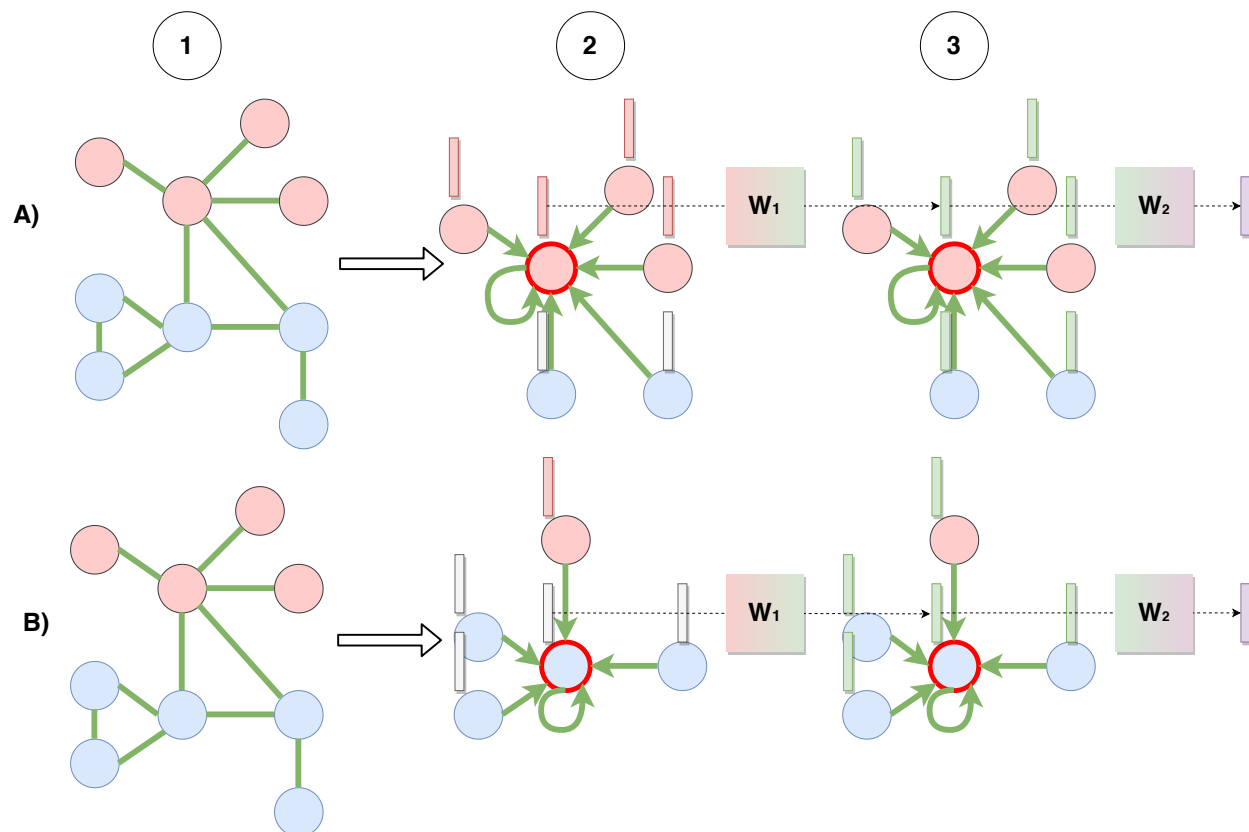


Figure 5.1: Illustration of a two-layer common GCN encoder. 1) The input is a multimodal network of disease nodes (red circles) and gene nodes (blue circles) with edges between and within node types (green lines). Figure **A)** shows the convolution for a single disease node, while Figure **B)** shows the convolution for a single gene node. 2) We train the same GCN encoder over disease nodes and gene nodes and all the edges, as indicated by the same weight matrix W_1 being used for both gene nodes and disease nodes. The encoder does not distinguish among disease-disease, gene-gene, and disease-gene edges. The input disease feature vectors (red bars) are what we call the Robinson Similarity Vectors (see Section 3.4.1), while the input gene feature vector (gray bars) are what we call Blank Vectors (see Section 3.4.2). 3) After each layer of convolution, feature information is propagated from the diseases to the genes, and vice versa. The layers of convolution are parameterized by the weight matrices W_1 and W_2 . The final embeddings (shown as purple bars) are the output of the encoder. *Note that we perform three layers of convolution in our actual implementation of the common GCN, but we show a two-layer common GCN in this figure, for ease of understanding.*

5.1.3 Multimodal GCN Encoder

In this section, we describe our main contribution: we train edge-type specific GCN encoders in the initial convolutional layers. Concretely, we train multiple layers of disease-specific GCN encoders to perform neighborhood aggregation over only the disease nodes and disease-disease edges. Separately, we train multiple layers of gene-specific GCN encoders to perform neighborhood aggregation over only the gene nodes and gene-gene edges in the multimodal graph. The separate gene and disease-specific convolutional layers transform the initially distinct input features spaces of genes and diseases into a common latent representation through multiple non-linear transformations. Once the disease and gene features have been transformed to a common latent feature space, we perform a convolution over the disease-gene edges in the multimodal network. This strategy allows information to flow from diseases to the neighborhood of its associated genes and vice versa.

An advantage of this approach is that the initial disease and gene features need not be in the same space and do not need to be of the same dimensionality. We use the Robinson Similarity vector for disease features and the STRING PPI Adjacency Vector for gene features (described in Section 3.4). We use two layers of 64-dimensional gene-specific GCN convolutions over only the gene nodes and the gene-gene edges and two layers of 64-dimensional disease-specific GCN convolutions over only disease nodes and the disease-disease edges, followed by a single layer of convolution over the disease-gene edges. A graphical representation of the multimodal GCN Encoder is shown in Figure 5.2.

5.1.4 Decoder

Once the disease and gene features are converted to the final embeddings by the encoder, we use a *decoder* to predict the probability of a disease-gene association. We experiment

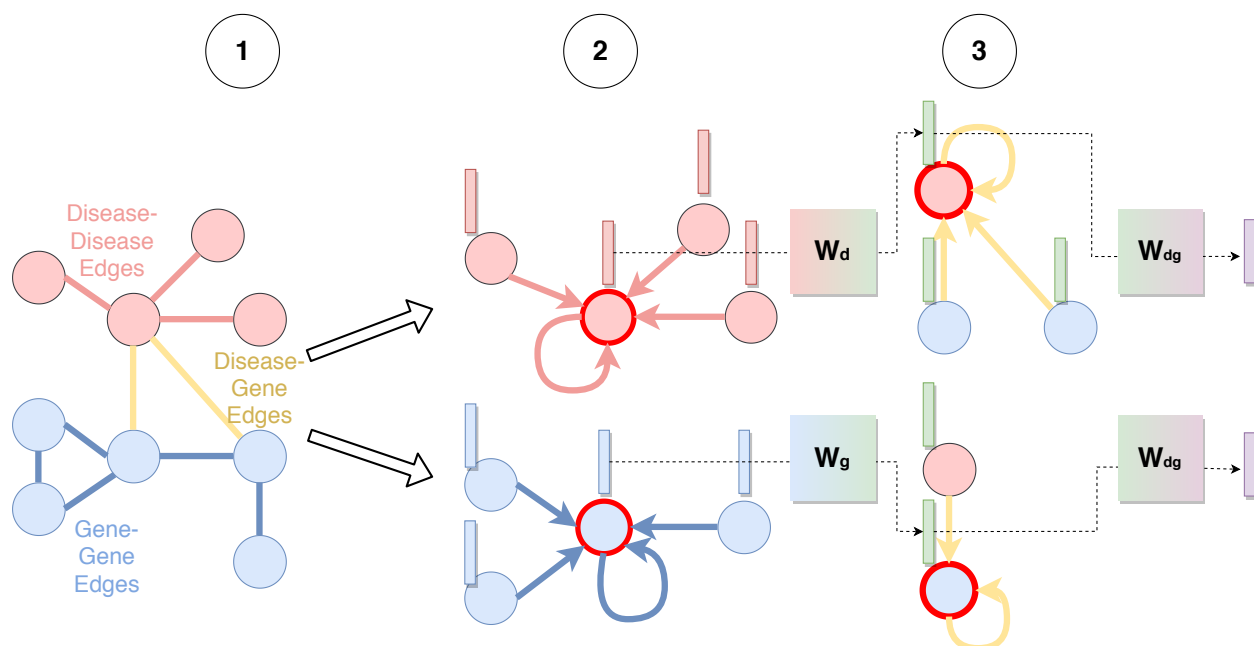


Figure 5.2: Illustration of the multimodal GCN encoder. 1) The input consists of a multimodal network of disease nodes (shown as red circles) and disease-disease edges (shown as red lines); gene nodes (shown as blue circles) and gene-gene edges (shown as blue lines), and disease-gene edges (shown as yellow lines). The initial disease feature vector (shown as red bars) is the Robinson Similarity Vector, and the initial gene feature vector (shown as blue bars) is the STRING PPI Adjacency Vector. 2) Separate GCN encoder layers are trained over the gene nodes, convolving over the gene-gene edges, parameterized by W_g and the disease nodes, convolving over the disease-disease edges, parameterized by W_d . The respective encoder layers transform the different feature spaces of the diseases and genes to a common latent representation (shown as green bars). *Note: this step consists of two convolutional layers over both diseases and genes, but only one layer is shown for simplicity* 3) The common latent representation allows us to perform convolution across the disease-gene edges (shown as yellow lines), using a disease-gene encoder layer, to get the final embedding of disease and gene nodes (shown as purple bars). This layer, parameterized by W_{dg} is the same for disease nodes and gene nodes.

with using two decoders, one which does not have any trainable parameters, called the *inner product decoder* and a parameterized decoder, called the *RESCAL decoder*.

Non-Parameterized Decoder: Inner Product Decoder. The inner product decoder simply computes the inner product of the final embedding of a disease d , and a gene g produced by the encoder, and then scales the inner product between 0 and 1 using the sigmoid function. Mathematically, given an embedding h_d for the disease node d and an embedding h_g for the gene node g , the probability of the disease-gene association represented by the edge (d, g) in the multimodal network is given by:

$$\Pr(d, g) = \sigma(h_d^T h_g),$$

where σ is the sigmoid function.

Parameterized Decoder: RESCAL Decoder. In this work, we also explore the effects of using a parameterized decoder, namely the RESCAL decoder [60]. Just as is the case for the inner product decoder, the RESCAL decoder computes the probability of a disease-gene association (d, g) by taking as input the final embedding h_d of the disease d and the encoding h_g of gene g . Here, a trainable parameter matrix R is used to convert the pair of embeddings to a scalar. The probability $\Pr(d, g)$ of the disease-gene edge (d, g) , is given as:

$$\Pr(d, g) = \sigma(h_d^T R h_g),$$

where σ is the sigmoid function.

5.1.5 Loss function

We train the encoders to output embeddings of the disease and gene nodes in the network so as to maximize the probability of a positive disease-gene association in the training set, while simultaneously minimizing the probability of the randomly generated “negative” disease-gene associations. The loss function we use to optimize the parameters of the encoders is *cross-entropy with negative sampling*, inspired by Mikolov *et al.* [52]. Given a set ν^+ of training positive disease-gene associations, and a set ν^- of training negative disease-gene associations, the loss function is defined as follows:

$$\ell(\nu^+, \nu^-) = \sum_{(d^+, g^+) \in \nu^+} -\log(\Pr(d^+, g^+)) + \sum_{(d^-, g^-) \in \nu^-} -\log(1 - \Pr(d^-, g^-))$$

5.2 Supervised Learning Approaches

We formulate the disease-gene prediction problem as a supervised learning task where the input features represent either a true positive or negative (disease, gene) pair. The output is a score representing the disease-gene association. We implement two supervised learning approaches. We describe the features, the methods, and the class labels below.

Features. For a disease-gene pair (d, g) , we create a feature vector by concatenating the disease feature vector ϕ_d with the gene feature vector ϕ_g (see Section 3.4).

Labels. We set the label of every true positive disease-gene pair in the training set (see Section 4.1) to be 1. For every true positive disease-gene pair, we sample negative pairs, in accordance with the negative sampling strategy described in Section 4.1.1.

5.2.1 Predictors

The classifiers we test are as follows:

1. **Logistic Regression:** As the simplest baseline, we train a single logistic regression classifier on our dataset of positive disease-gene associations and randomly sampled negatives.
2. **Multi-layer Perceptron (MLP):** In this approach, we use the same inputs and labels as the Logistic Regression approach, but we use a feed forward MLP as the classifier. The MLP has three layers, with 64 neurons in each layer, and a single neuron in the final layer, which emits a scalar representing the probability of the input disease-gene association.

The parameters for the predictors are updated during training using the Adam optimizer [31].

5.3 Cardigan

Cardigan is an appropriate choice to benchmark our approach, since it allows predictions to be made for diseases with no known genes, using a measure of disease similarity to “transfer” knowledge from diseases with known gene associations, to those with no such known associations. This method is based on the assumption that diseases which are similar according to Cardigan’s measure of disease similarity will have common or interacting associated genes. It has demonstrated better results for this task than multiple well-known disease-gene prediction algorithms, including ProDiGe [56], PRINCE [86], and DIAMOnD [19], methods which themselves outperform or are competitive with a number of disease-gene prediction algorithms like BiRW [94] and Endeavour [4]. This makes Cardigan a formidable baseline

to benchmark against our predictors.

We describe the Cardigan network-propagation algorithm in detail below. The approach expects the following inputs:

- A PPI network $G = (V, E, W)$, with a set of vertices V of size v , a set of edges E of size e , and the $v \times v$ matrix of edge weights W , where if $(i, j) \in E$, W_{ij} corresponds to the weight of the edge between node i and node j .
- A $d \times d$ disease similarity matrix D , where the cell in row i and column j represent some measure of similarity between a pair of diseases d_i and d_j .
- A query disease d_q .
- A set of disease-gene associations S_{dg} .

Cardigan uses these inputs to assign a score $F(i)$ to each node i in G , corresponding to the likelihood of its association with d_q .

Let δ be the degree matrix of the graph. δ is a $v \times v$ diagonal matrix, where $\delta_{ii} = \sqrt{\sum_{k:(i,k) \in E} W_{ik}}$. σ is the sigmoid function, and μ is a multiplicative factor controlling the contribution of the second half of the cost function. Let $Y : V \rightarrow [0, 1]$ be the initial value function, assigning an initial score to each node in V . $Y(i)$ is the initial value assigned to node i in G , according to the following rules:

$$Y(i) = \begin{cases} 1 & \text{if the association } (d_q, i) \in S_{dg} \\ 0 & \text{if } i \text{ is not associated with any diseases} \\ \max_{r:(r,i) \in S_{dg}} (\sigma(D_{qr})) & \text{for all other genes } i \end{cases}$$

Using $F : V \rightarrow [0, 1]$ to denote the scoring function, where $F(i)$ is the final score of gene i .

The cost that Cardigan optimizes is the following:

$$C(F) = \frac{1}{2} \left(\sum_{i,j \in V} W_{ij} \left(\left\| \frac{1}{\delta_{ii}} F(i) - \frac{1}{\delta_{jj}} F(j) \right\|^2 \right) + \mu \sum_{i \in V} \|F(i) - Y(i)\|^2 \right) \quad (5.3)$$

The minimized cost function has a closed form solution, which depends only on the initial labelling vector Y , μ , and the properties of the network G . Given the $v \times v$ identity matrix I , the minimized value of F for all genes in V is given by a $v \times 1$ vector, $\bar{F}$, where the a^{th} row in $\bar{F}$ corresponds to the minimized score of the gene corresponding to the a^{th} row/column in W :

$$\bar{F} = \frac{\mu}{1 + \mu} \left(I - \frac{1}{\mu} \delta^{-1/2} W \delta^{-1/2} \right)^{-1} Y \quad (5.4)$$

5.3.1 Modifications to Cardigan

In order to make the evaluation and comparison fair, we make a few modifications to the original Cardigan algorithm, so as to allow it to run on the same input data and the same cross-validation procedure as our methods.

Disease similarity matrix. The Caniza disease similarity matrix [11] used in the original Cardigan algorithm is generated using the Medical Subject Heading (MeSH) hierarchy. The authors obtain the publications associated with each disease in consideration through PubMed, which is an online bibliographic biomedical database. Each PubMed publication is indexed with a set of MeSH terms, which are a thesaurus of medical descriptors, organized in a hierarchical manner. These MeSH terms are associated with 16 categories, including anatomy, phenomena and processes, and health care, and contain disease phenotype infor-

mation (along with other information relevant to the disease). The similarity between two diseases is found based on the set of the MeSH terms mapped to a disease, and uses the MeSH hierarchy.

In our work, we use the HPO for disease-phenotype associations, and not disease-MeSH term associations, which cover broader categories of disease properties. Moreover, we use both OMIM and Orphanet diseases for our methods, while the original Caniza similarity matrix was built from text mining of MeSH terms for only OMIM diseases. The parsing of the PubMed publications associated with OMIM diseases which was used in the Caniza matrix generation was not available at the time of this work. This parsing was required in order to find the PubMed publications associated with a disease. These publications would be used to obtain the associated MeSH terms, and finally to generate the Caniza similarity matrix for associations between OMIM diseases and PubMed publications. Moreover, data about associated PubMed publications was also not easy to obtain for Orphanet diseases.

For these reasons, instead of the original Caniza similarity matrix, we use the HPO Robinson disease similarity matrix [75] (described in Section 3.4.1) as the disease similarity matrix for Cardigan.

Cross Validation. Cardigan is designed to predict associations only for a single disease at a time. In the original paper, the evaluation is done in a leave-one-disease-out manner, where only the gene associations for the query disease are held out during evaluation. However, we perform a 70-20-10 train-test-validation split of the data for our models (see Chapter 4). Hence when we run the Cardigan algorithm for a test disease, we hold out all the disease-gene associations in the test set, using only training and validation disease-gene associations as seeds for information propagation.

Inputs. We run Cardigan on the same unweighted, undirected, preprocessed STRING interactome, diseases, and disease-gene associations as described in Section 3.1.

Scoring Positive and Negative Test Associations. Given a query disease, Cardigan produces a ranked list of **all** genes in the interactome, in decreasing order of probability of association. However, our approaches score positive and negative disease-gene associations in a test set with a fixed ratio of positives to negatives. Thus, we modify the output produced by Cardigan, which contains a ranked list of all V interactome nodes for each test disease. For every positive or negative disease-gene association (d, g) in the test set, we find the position P_g of g in the ranked list of all V genes, for d . We assign the score $1 - \frac{P_g - 1}{N - 1}$ to the association (d, g) . For example, g is the top ranked (i.e. first) gene in the list for d , association (d, g) gets a score of 1, and if it is the lowest ranked (i.e. N^{th}) gene in the list, it gets a score of 0.

Chapter 6

Results

In this chapter, we describe the results obtained on the methods we have described in Chapter 5 evaluated using the measures defined in Chapter 4. We also provide empirical evidence for some of the design choices we made, including the features and model architecture considerations. We show results both on the *per-association cross validation*, and *per-disease cross validation*. We present results for the *ranking based evaluation* as well as the *disease specific evaluation*. For each analysis, we study how the performance of the predictor varies as the distribution of negatives to positives in the dataset becomes more unbalanced (from 1:1 to 5:1 to 10:1). We show the results for the supervised learning approaches (logistic regression and multi-layer perceptron) in shades of blue, for Cardigan in green, and for the GCN approaches in shades of orange. The performance for a random predictor is indicated by a single black line.

6.1 Per-Association Cross Validation

In figure 6.1, we show the results on the test set, after splitting disease-gene associations uniformly at random into training (70%), test (20%), and validation (10%) sets, over five iterations of repeated random subsampling. We show the scores for the area under the precision recall curve (AUPRC) in Figure 6.1a and the early precision (EP) in Figure 6.1b, as described in Section 4.2. We show the trends of performance across all the algorithms,

across the negative:positive ratios of 1:1, 5:1 and 10:1, for the per-association cross validation. The performance of a random classifier is shown as a single black line across the figure.

In Figure 6.1c, we explore the model performance, when evaluated on a per-disease basis. We show the trends of the proportion of true positive disease-genes which are scored higher than all randomly sampled negative genes for a disease. We vary the negative:positive ratio between 1:1, 5:1, and 10:1, and show results across all test diseases, using per-association cross validation.

First, we observe that the performance for all methods decreases (as expected) when we increase the negative:positive ratio. We note that Logistic Regression performs roughly at par with a random predictor. Given the simplicity of the predictor, and the fact that we are training a single model over all diseases, this result may not be surprising.

We also see that both the GCN approaches are consistently better than the MLP by a fairly large margin. This shows that graph machine learning based methods have a performance improvement over naive supervised learning approaches with a similar number of parameters, and the same or similar input features. The performance of the MLP approach drops sharply as the negative to positive ratio in the dataset increases, while both Cardigan and the GCN approaches show only a gradual performance decrease as this ratio increases.

We observe that the Multimodal GCN shows an improvement over the Common GCN approach, despite having an equivalent number of parameters, showing that the separate convolutional encoders over different edge types improve performance. We see that Cardigan has slightly better AUPRC than the next best method, the Multimodal GCN, when the negative:positive imbalance in the dataset increases. However, both the Common GCN encoder and the Multimodal GCN encoder demonstrate better Early Precision than Cardigan across the entire range of negative to positive ratios.

This shift in trends of the relative performance of the methods, when comparing AUPRC and Early Precision means that the GCN approaches make a larger proportion of true positive predictions for high scoring disease-gene associations (i.e., early in the ranked list of predictions) than Cardigan, whereas Cardigan is relatively better at scoring true positive predictions ahead of negatives for lower scoring disease-gene associations (i.e., later in the ranked list of positive and negative disease-gene associations).

We also see that the Multimodal GCN outperforms all other methods over every negative:positive ratio for the disease-specific evaluation. This result shows that across all diseases, the Multimodal GCN assigns the true-positive disease genes higher scores than all the negative genes for that disease a higher percentage of times.

6.2 Per-Disease Cross Validation

Here, we show the results on the test set, after randomly partitioning diseases themselves into training (70%), test (20%), and validation (10%) sets. We again perform five iterations of repeated random subsampling as above, this time over the diseases. In each iteration, all the genes associated with a disease are in exactly one of the three sets. We randomly sample negative disease-gene associations for each set in the proportions described above. For each iteration, the positive and negative associations in the training, test, and validation set are consistent across each algorithm. In this way, we simulate the case where none of the diseases in the test set have known disease-gene associations in the training or the validation set. This way of evaluation aligns closely with the problem the thesis tries to tackle, which is that of finding gene associations for diseases with no known gene associations, and as such, forms our most important set of results.

We plot the scores for the area under the precision recall curve (AUPRC), and the early pre-

cision (EP), as described in Section 4.2. The trends of performance across all the algorithms, across the negative:positive ratios of 1:1, 5:1 and 10:1, for the per-disease cross validation are shown in Figure 6.2.

Here, we evaluate the model performance on a per-disease basis. We show the trends of the proportion of true positive disease-genes which are scored higher than all randomly sampled negative genes for a disease. We vary the negative:positive ratio between 1:1, 5:1, and 10:1, and show results across all test diseases, using per-disease cross validation, in Figure 6.2c.

We compare the results for the per-disease cross validation with those of the per-association cross validation discussed in Section 6.1, noting differences in the trends we observe.

We observe that the trends in performance for the disease specific evaluation measure (Figure 6.2c) remain roughly the same as they were in the per-association cross validation (Figure 6.1c), with the Multimodal GCN performing the best among all methods, followed by Cardigan. The Logistic Regression classifier performs worse than random. However, we see that the average performance for all the other methods decreases for per-disease cross validation, when compared with the per-association cross validation. This trend aligns with our expectations, since it is understood that it is more challenging to predict genes for a disease with no known associations. We make this property hold true for all the diseases in the test set, through our method of cross validation. We do notice, however, that the drop in performance of Cardigan is relatively sharper than that of the Multimodal GCN or the Common GCN approach. While the differences between Cardigan and the Multimodal GCN were small in the disease specific evaluation in per-association cross validation case, they are pronounced in this case.

For the AUPRC, the trends in performance among the algorithms for the per-disease cross validation (Figure 6.2a) are again the same as the per-association case (Figure 6.1a). The

performance for the MLP and for Cardigan decrease for the per-disease validation in comparison to the per-association evaluation. In contrast, the performance for the two GCN approaches remains essentially the same in both evaluations.

For the EP measure, we see again that the performance for the GCN based approaches remains almost the same in the per-disease cross validation (Figure 6.2b) as the per-association (Figure 6.1b) case. The performance for the MLP decreases compared to the per-association case. Strangely enough, the performance for Cardigan is considerably higher for the EP measure, for the per-disease cross validation, compared to the per-association case. In fact, it is higher than even the Multimodal GCN for a negative:positive ratio of 10:1. Further analysis is required to understand why this is the case.

6.3 Effects of Design Choices

Here, we explore the empirical effects of different design choices on the performance of our methods. In particular, we explore the effects of using different feature vectors as inputs for the Multimodal GCN and the Common GCN. We also explore the effects of using different numbers of nearest neighbors to construct the disease similarity graph, and the effects of using the RESCAL Decoder instead of the Inner Product Decoder in our Multimodal GCN approach.

6.3.1 Effect of Number of Nearest Neighbors in Disease-Similarity Graph

Recall that the multimodal graph that we use in the link-prediction approaches contains a disease-similarity network. The feature vector for each disease node in this network is the

256-dimensional Robinson Similarity Vector. Each disease in the network is connected by an unweighted, undirected edge to the five diseases whose features are closest to its own feature vector, measured in terms of Euclidean distance.

Since we are presented with a choice of how many nearest neighbors to connect each disease with, we probe the effects of varying the number of neighbors in the graph between 3,5,7, and 10, in Figure 6.3. We demonstrate the results on the ranking based evaluation, on the Multimodal GCN model, with a negative:positive ratio of 10:1, with the per-disease cross validation. We see that the performance for the ranking based evaluation measures noticeably drops when the number of neighbors is reduced or increased even slightly. Thus, it seems that five is an appropriate choice for the number of nearest neighbors in the disease-similarity graph.

6.3.2 Effect of Different Gene Feature Vectors

Here we experiment with using different gene features as the initial node features over the multimodal graph for both the Common GCN and the Multimodal GCN. In each case, the initial disease features are the same, i.e., the 256-dimensional Robinson Similarity vector. We instantiate the GCN models with the following gene features:

- **Common GCN Blank Vector:** Common GCN with the Blank Vector as gene feature (i.e., the features used in Section 6.1, and Section 6.2 for this variant).
- **Common GCN PCA:** Common GCN with the STRING PPI adjacency vector as gene features.
- **Multimodal GCN PCA:** Multimodal GCN with the STRING PPI adjacency vector as gene features (i.e., the features used in Section 6.1, and Section 6.2 for this variant).

- **Multimodal GCN Blank Vector:** Multimodal GCN with the Blank Vector as gene features.

We compare performance for the ranking based evaluation on a negative:positive ratio of 10:1, using the per-disease cross validation. The results are shown in Figure 6.4.

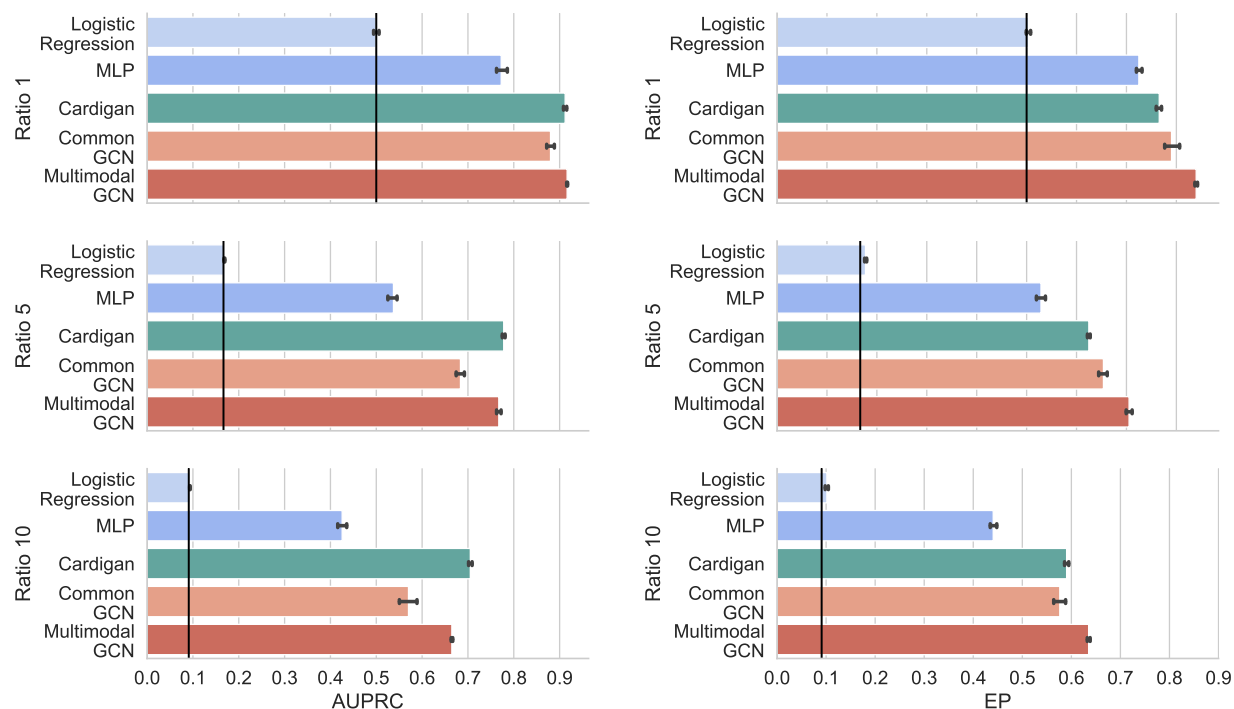
We find that our motivations for using the multimodal convolutional approach are validated by this analysis. We see that the Common GCN PCA performs worse than Common GCN Blank Vector. Since the Common GCN approach trains a single encoder over all edge types and over all nodes, the different feature spaces of the genes and diseases cannot effectively be reconciled using a single encoder over the distinct feature spaces of diseases and genes. That is why *using no gene feature information at all*, (i.e., using a Blank Vector for gene features) actually performs better than having gene features in this case. However, if we perform the convolution over genes and diseases in a principled way, i.e., by using separate encoders over gene and disease nodes, and performing convolutions over disease-gene edges only **after** the encodings for disease and genes are in a common latent space, then using informative gene features actually improves performance over using 0-vectors. We also observe that the Multimodal GCN Blank Vector approach, despite performing worse than the Multimodal GCN PCA approach, still performs better than the Common GCN 0 approach, showing that doing edge-type specific convolutions also improves performance in the case where the disease and gene features are reconcilable with each other.

6.3.3 Effect of Using the RESCAL Decoder

The results obtained on the ranking task, when comparing the inner product decoder and the RESCAL decoder are shown in Figure 6.5. We use the Multimodal GCN encoder, and a negative:positive ratio of 10:1, on the per-disease cross validation. We see that parameterized

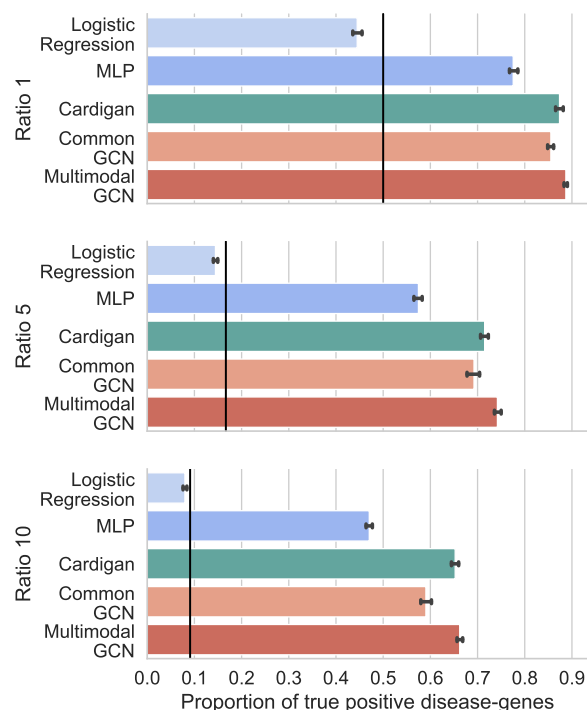
decoding decreases the performance on the ranking task.

Figure 6.1: AUPRC, early precision and disease specific evaluation results, for all algorithms for per-association cross validation and negative:positive ratios of 1:1, 5:1, and 10:1.



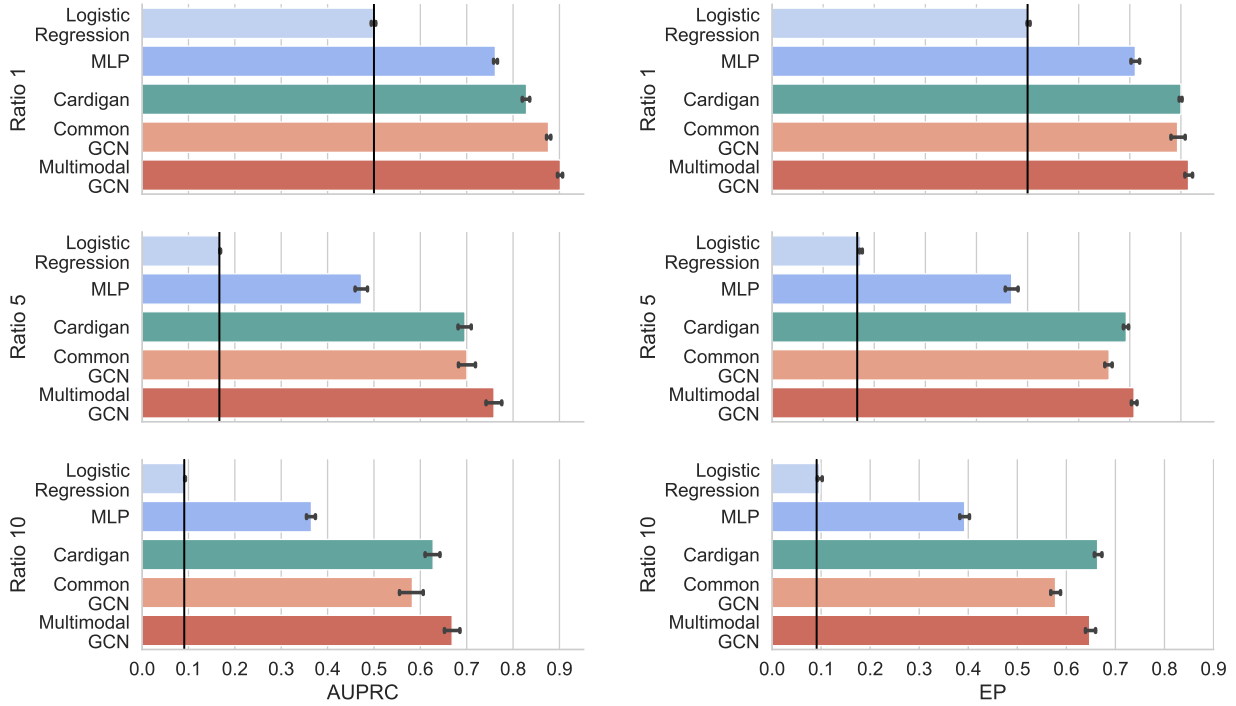
(a) AUPRC evaluation with per-association cross validation

(b) EP evaluation with per-association cross validation



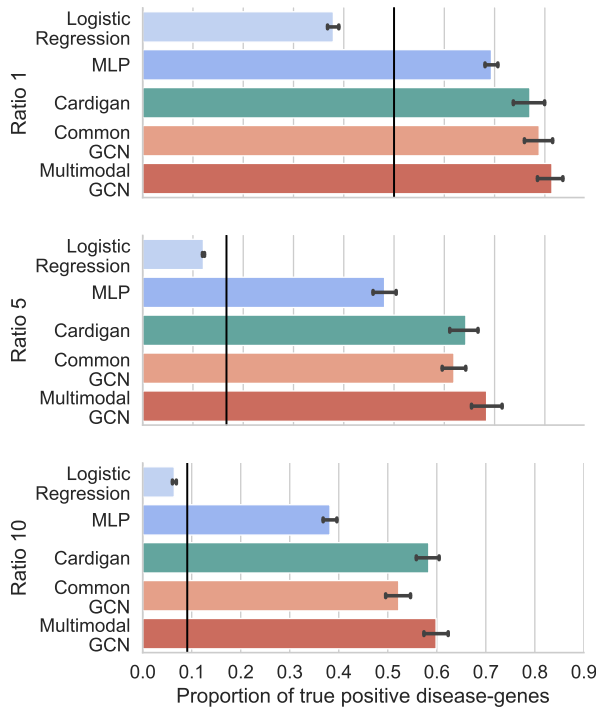
(c) Disease specific evaluation with per-association cross validation

Figure 6.2: AUPRC, early precision and disease specific evaluation results, for all algorithms for per-disease cross validation and negative:positive ratios of 1:1, 5:1, and 10:1.



(a) AUPRC evaluation with per-disease cross validation

(b) EP evaluation with per-disease cross validation



(c) Disease specific evaluation with per-disease cross validation

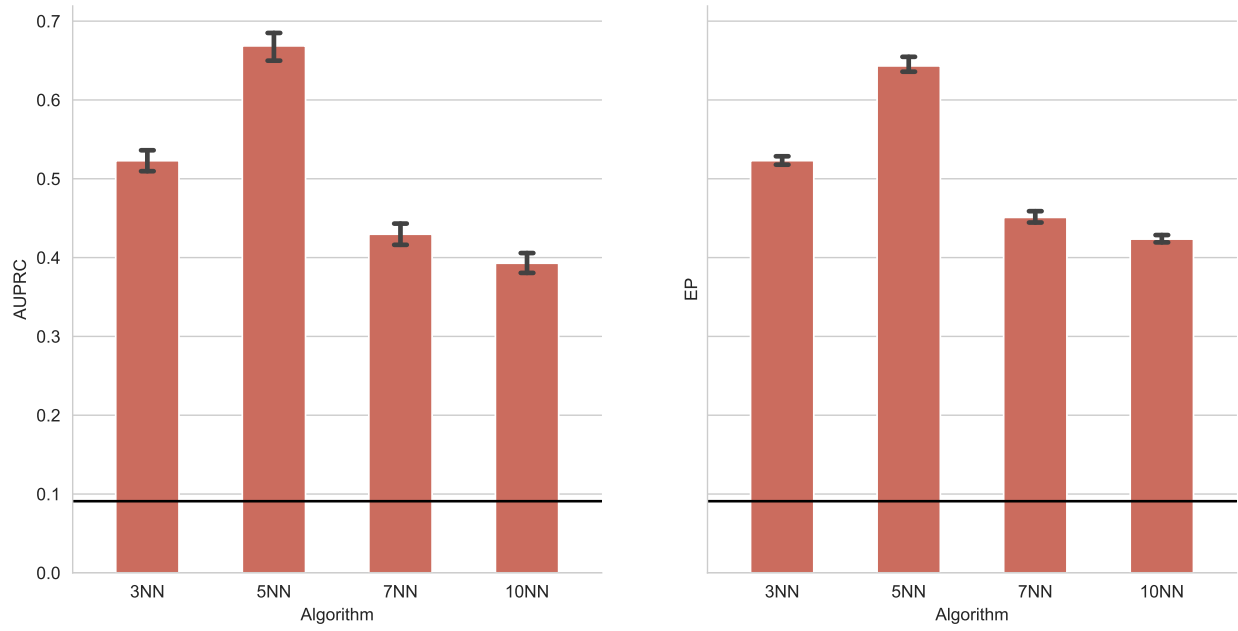


Figure 6.3: Values of (a) AUPRC and (b) early precision for different values of nearest neighbors (3,5,7,10) in the disease similarity graph. The results are for the Multimodal GCN model with a negative:positive ratio of 10:1 on per-disease cross validation.

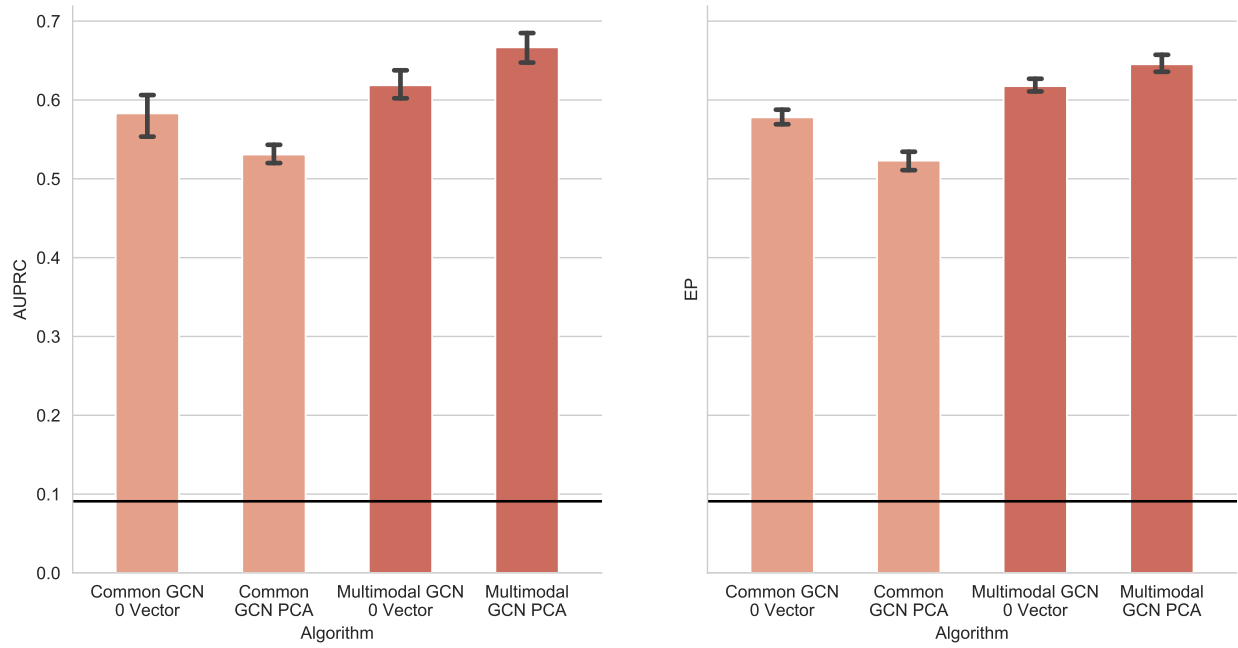


Figure 6.4: Trends of the AUPRC and Early Precision across different types of inputs to the GCN models. The results are shown for the Common GCN model, and the Multimodal GCN model with a negative:positive ratio of 10:1, on the per-disease cross validation.

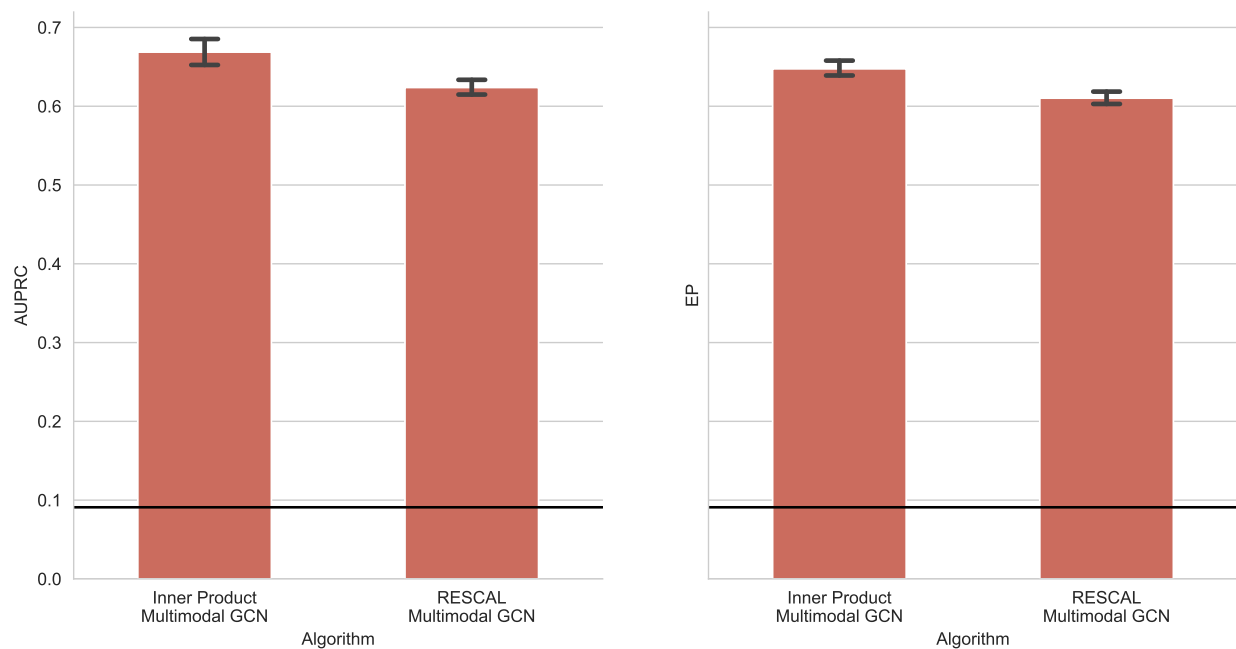


Figure 6.5: Comparison of AUPRC and early precision between the inner product decoder and the RESCAL decoder. The results are shown for the Multimodal GCN model with a negative:positive ratio of 10:1, on the per-disease cross validation.

Chapter 7

Analysis and Discussion

In this chapter, we delve into some analysis of the final layer embeddings produced by the GCN models. In Section 7.1, we explore the embeddings produced by the Multimodal GCN for disease and gene nodes, contrasted with the raw disease and gene features. We also compare the embeddings produced by the Multimodal GCN, with those of the Common GCN. In Section 7.2 we analyze the concordance of results across different methods, exploring whether the set of diseases for which associations are correctly predicted, and the set of diseases for which the associations are incorrectly predicted, are consistent across algorithms.

7.1 Embedding Analysis

Here, we analyze the final layer embeddings produced by the Multimodal GCN for diseases and genes, and compare them to the raw disease and gene features used as input. The results are shown over five iterations of repeated random subsampling, and per-disease cross validation. The raw features for the Multimodal GCN are the Robinson Similarity Vector for diseases and the STRING PPI Adjacency Vector for genes, and thus represent different feature spaces. In this work, we claim that the Multimodal GCN brings these initially distinct input features spaces to a common latent feature space, and is moreover incentivized to make the final embedding of a disease as similar as possible to its associated genes (See Section 5.1.3), through the loss function (See Section 5.1.5).

To experimentally verify this claim, we plot the distribution of the cosine similarity of the final embeddings of each true positive disease gene pair in our test set. We also plot the distribution of the cosine similarities of the initial gene and disease feature vectors for these disease gene pairs, for comparison.

Given two embedding vectors h_1 and h_2 , and the norm function $\|\cdot\|$, the cosine similarity of h_1 and h_2 is given as:

$\text{sim}_{\text{cosine}}(h_1, h_2) = \frac{h_1 \cdot h_2}{\|h_1\| \|h_2\|}$. The cosine similarity measures the cosine of the angle between two vectors, and ranges between -1 (the vectors are at an angle of 180° w.r.t. each other) and 1 (the vectors are at an angle of 0° w.r.t. each other). We would expect the cosine similarity of the final embeddings of a disease gene pair to be higher than that of the initial raw feature vectors for the pair. We find that this is the case, as shown in Figure 7.1. The cosine similarity for the final embeddings of each test disease gene pair is on average considerably higher than that of their raw feature vectors. In fact, we see that the cosine similarity distribution for the raw features is close to zero on average, which is reasonable since the features represent entirely different quantities.

Despite being in the same feature space, we would expect the cosine similarity of the final embeddings of negative disease-gene associations in the test set to be much lower than that of positively associated test disease gene pairs. To verify this, we compare the distributions of the cosine similarities of final embeddings produced by the Multimodal GCN encoder, for positive and negative test disease-gene associations in Figure 7.2, and find that this is indeed the case. The cosine similarity of negative test disease-gene pairs is on average significantly lower than that of positive test disease-gene pairs, and is in fact centered at 0.

Another claim we make in this work is that a Common GCN Encoder is unable to successfully perform neighborhood aggregation in the case where the gene and disease feature vector

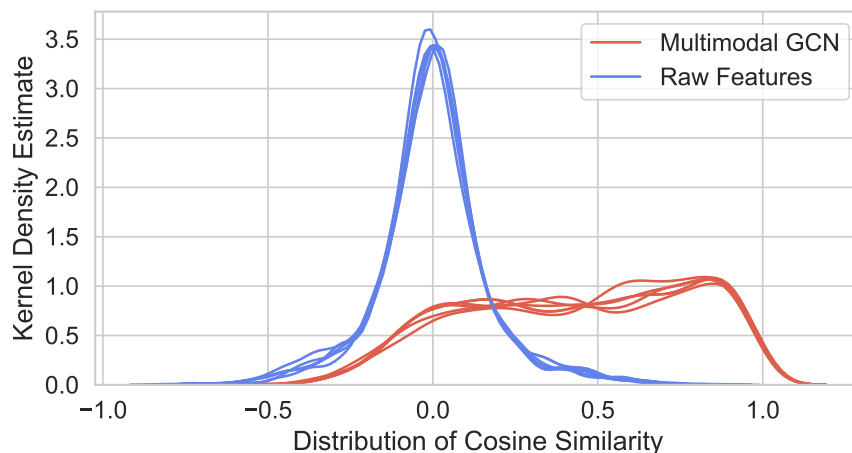


Figure 7.1: Comparison of the distribution of cosine similarities of the final embeddings of true positive test disease gene pairs, and the distribution of cosine similarities of the raw features for true positive disease gene pairs. Results are for the per-disease cross validation, with a negative:positive ratio of 1:1, and the distributions are shown for all five iterations of repeated random subsampling.

spaces are different (See Section 5.1.2). This is a major motivation for using the Multimodal GCN encoder. We explored whether we could explore some properties of the final embeddings produced by the Common GCN, compared to those of the Multimodal GCN, when the input features are the Robinson Similarity Vector for diseases and the STRING PPI Adjacency Vector for genes.

Plotting the distribution of inner products of positive and negative disease gene pairs for the Multimodal GCN case, as opposed to the Common GCN case in Figure 7.3, we see that using a Multimodal GCN “pushes” the inner product of true positive disease gene pairs to a larger value on average, as compared to the Common GCN. Since the decoder is an inner product decoder, this means that the true-positive disease-gene associations have a higher score in the Multimodal GCN case. Moreover, we see that the distribution of inner products for negative test disease gene edges is quite similar both for the Common GCN and the Multimodal GCN. This shows us that the boost in performance from using a Multimodal GCN comes from the greater confidence assigned to true positive test disease-gene associations, and not

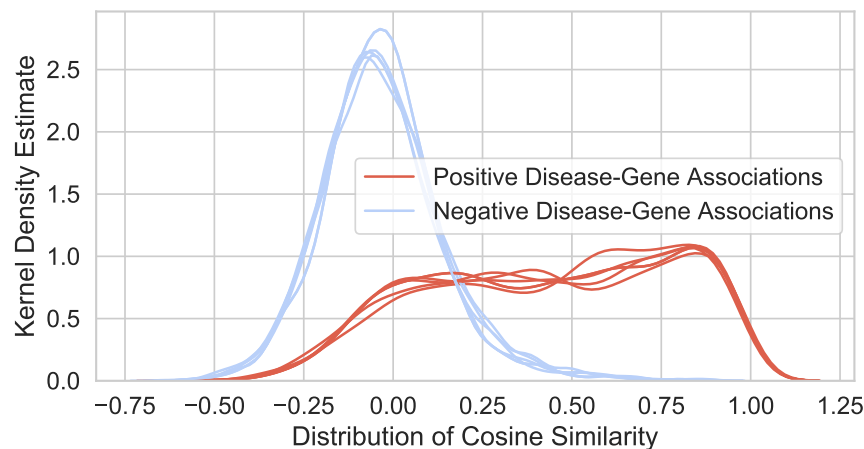


Figure 7.2: Comparison of the distribution of cosine similarities of the final layer embeddings of true positive test disease gene pairs, and the distribution of cosine similarities of embeddings of negative test disease gene pairs. Results are for the per-disease cross validation, with a negative:positive ratio of 1:1, and the distributions are shown for all five iterations of repeated random subsampling.

from greater ability to weed out test negative disease-gene associations.

This analysis helps us understand why having separate edge-specific convolutions as opposed to common convolutions over all edges improves performance when the disease and gene feature spaces are distinct (See Common GCN PCA and Multimodal GCN PCA in Figure 6.4).

7.2 Prediction Consistency Analysis

Here, we explore the extent to which the set of test diseases for which the Multimodal GCN approaches cannot find true positive gene associations intersects with the the set of test diseases for which Cardigan cannot find true positive gene associations. Put another way, we wanted to answer the question, “*What is the overlap between the diseases for which Cardigan finds it difficult to make gene predictions, and those for which the Multimodal GCN*

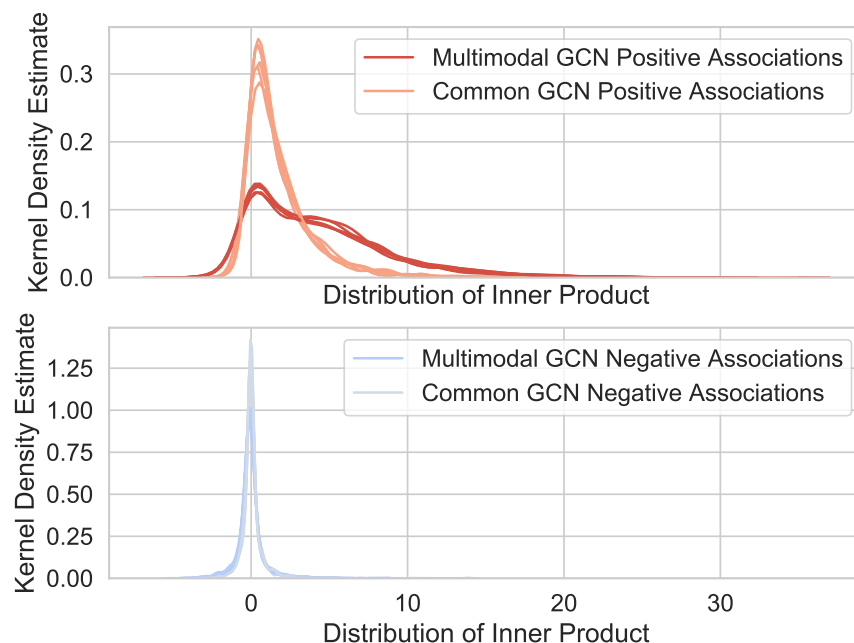


Figure 7.3: Comparison of the distribution of inner products of final layer embeddings for true positive (top) test disease gene pairs, and negative (bottom) test disease gene pairs for the Common GCN, as compared to the Multimodal GCN. Results are for the per-disease cross validation, with a negative:positive ratio of 1:1, and the distributions are shown for all five iterations of repeated random subsampling.

finds it difficult to make gene predictions?”

To answer this question, we defined two sets of true positive test disease-gene associations, **high scoring associations**, and **top ranked associations**. A true positive disease-gene association is called a *high scoring* association, if it is among the top scoring n^+ associations in the ranked list of all test disease-gene associations, where n^+ is the number of true positive test disease-gene associations. A true positive disease-gene association in the test set is called a *top ranked* association, if it is scored higher than all negative gene associations in the test set, for the same disease.

To explore this, we divided the set of test diseases into three categories, which we call *disease classes*, namely **easily associable**, **moderately associable**, and **hard to associate**.

These terms are defined as follows:

- **Easily associable:** These are the diseases for which some test disease-gene association is considered easy to recover. They are diseases where **at least one** true positive test disease-gene association is both high scoring and top ranked.
- **Hard to associate:** These are the diseases for which all disease-gene associations are considered hard to recover. They are the diseases where **none** of the true positive test disease-gene associations are either high scoring or top ranked.
- **Moderately associable:** These are the diseases for which disease-gene associations are considered moderate to recover. These are diseases which are neither easily associable, nor hard to associate.

We obtain a set of easily associable, moderately associable and hard to associate diseases, for the per-disease cross validation, over five iterations of repeated random subsampling, and a negative:positive ratio of 1:1. We plot a heatmap of the average size (over each iteration of random subsampling) of the intersection of each of the three disease classes for the Multimodal GCN, as compared with Cardigan, in Figure 7.4. We see from this figure that the set of hard to associate diseases according to the Multimodal GCN is very evenly spread out over all three disease classes for Cardigan (see the third column), and vice versa (see the third row). This means that there is no consistent definition of what constitutes a “hard to associate” disease, across both algorithms.

Since there is a large disparity in the size of the three disease classes, (i.e. the number of easily associable diseases is a lot higher than the number of diseases in the other two disease classes), we row-normalize the matrix from the previous figure (Figure 7.4). The heatmap of the row-normalized matrix is shown in Figure 7.5. In the row-normalized heatmap, each

entry (row x , column y) can be interpreted as “*the proportion of diseases which belong to disease class x for Cardigan, but disease class y for the Multimodal GCN*”. We see that this figure shows higher values in the first column and third row. This is an indirect signifier that the Multimodal GCN approach performs better overall than Cardigan. From this figure, we see that a lot of the performance improvement for the Multimodal GCN over Cardigan is because a large number of moderately associable diseases for Cardigan, are easily associable for the Multimodal GCN (see the second row, first column). The fact that both algorithm types find very different sets of diseases hard to associate, leads us to conjecture that an ensemble approach combining the network propagation of Cardigan, and a multimodal graph machine learning approach, might outperform both individual approaches for this task.

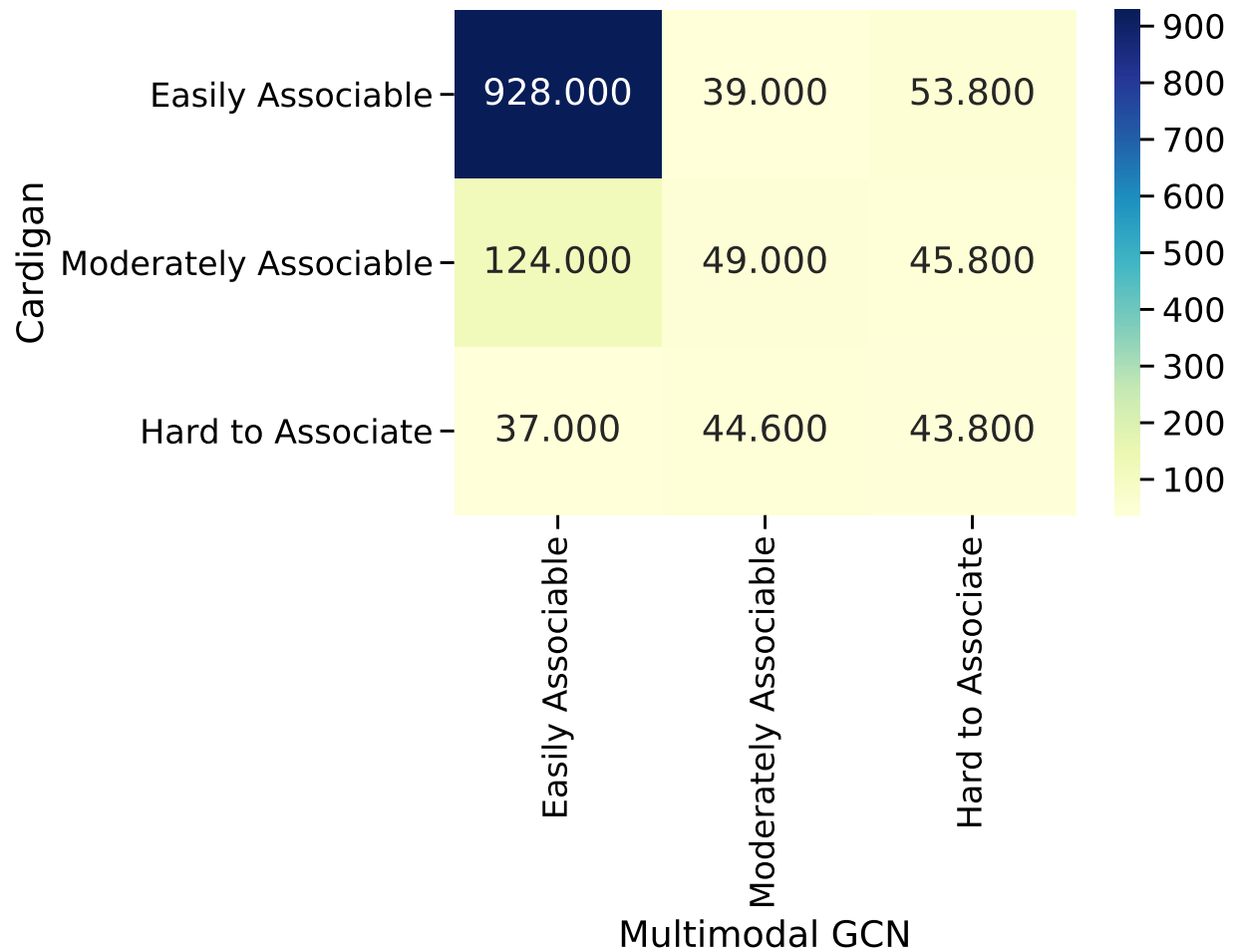


Figure 7.4: Heatmap representing average size of the intersection of the sets of test diseases in the three disease classes for Cardigan, compared to the Multimodal GCN. Results are for the per-disease cross validation, with a negative:positive ratio of 1:1, averaged over all five iterations of repeated random subsampling.

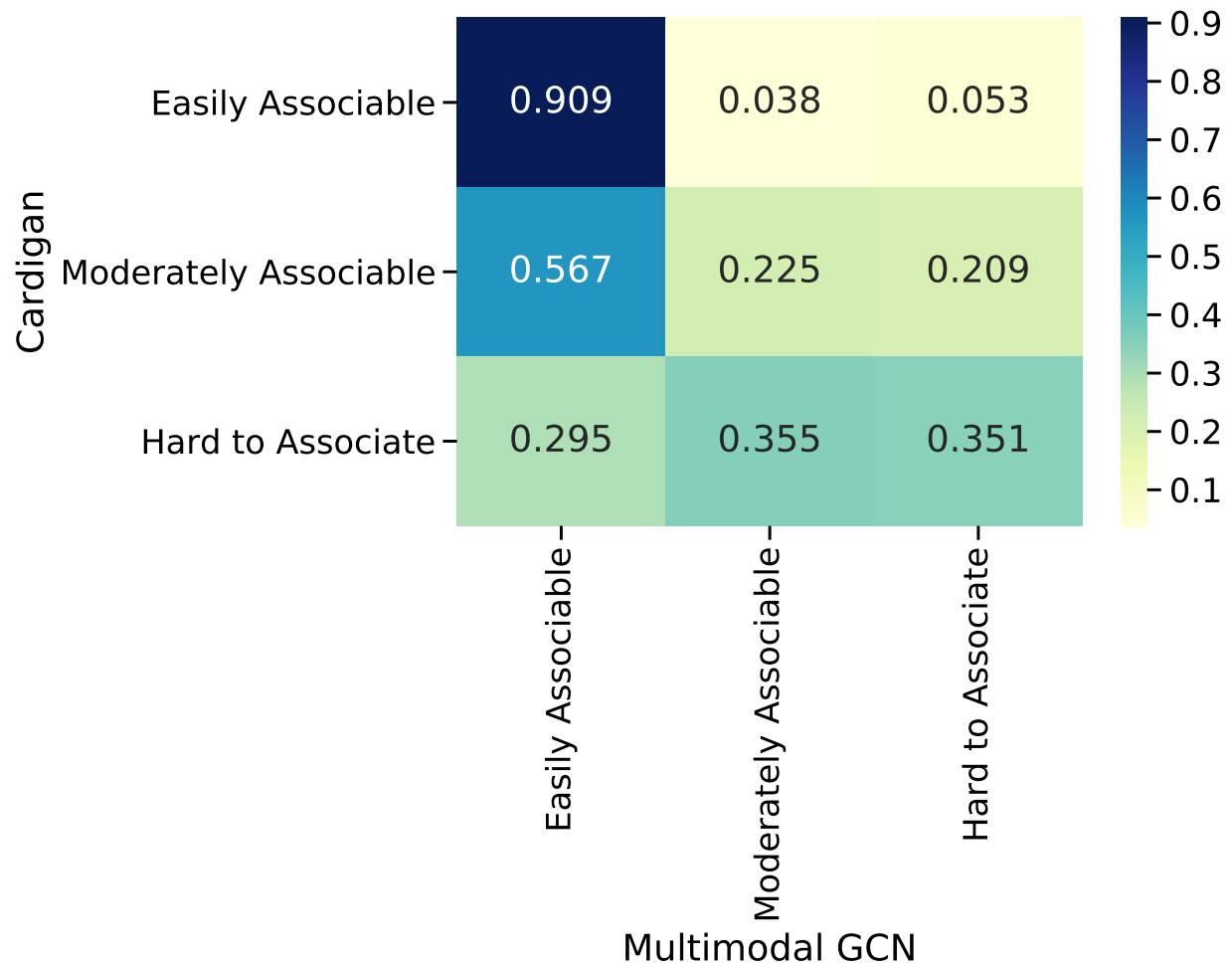


Figure 7.5: Heatmap where the cell in the x^{th} row and y^{th} column represents the proportion of diseases which belong to disease class x for Cardigan, but disease class y for the Multimodal GCN. Results are for the per-disease cross validation, with a negative:positive ratio of 1:1, averaged over all five iterations of repeated random subsampling.

Chapter 8

Conclusion and Future Work

We began this work by noting that finding the genes associated with rare genetic diseases is a problem of great importance to society (see Section 1.1). We observed that computational tools which can predict the genes associated with rare diseases can be vital aids for this task (see Section 1.4), since non-computational methods (GWAS and Linkage Analysis) have drawbacks when it comes to predicting genes in the case of rare diseases (see Section 1.3). However, we noted that most computational methods for predicting the genes associated with rare diseases cannot make associations for diseases with no already known gene associations; a category under which a huge proportion of rare genetic diseases fall (see Section 1.1.2). We saw that graph machine learning, and graph convolutional networks are gaining great popularity in bioinformatics, and have shown potential for a diverse array of predictive tasks (see Section 1.6). Driven by these motivating observations, we set out to develop a graph convolutional approach which can make predictions for diseases with no known associated genes.

We developed two approaches, the Common GCN (see Section 5.1.2), and the Multimodal GCN (see Section 5.1.3). We benchmarked these approaches against two supervised learning approaches, MLP and Logistic Regression (see Section 5.2), and a recently developed network propagation approach which has been developed to make predictions for diseases with no known associated genes, namely Cardigan (see Section 5.3).

We saw that the GCN based approaches outperform supervised learning approaches on ev-

every single evaluation measure, in Section 6.1 and Section 6.2, highlighting the effectiveness of graph machine learning over other machine learning methods. We see that our Multimodal GCN outperforms the Common GCN over all evaluation measures in Section 6.1 and Section 6.2, showing that training separate convolutional encoders over the different edge types improves performance in these measures. Vitally, we see that it is the multimodal convolutional encoders, which enable the Multimodal GCN to outperform Cardigan, while the Common GCN shows inferior performance to Cardigan across most evaluation measures.

We explored the properties of the embeddings produced by the Multimodal GCN, in Chapter 7. We see that the Multimodal GCN brings the embeddings of positively associated genes and diseases together, while keeping the embeddings of negatively associated genes and diseases far apart, as evidenced by the cosine similarity distributions in Section 7.1. We also see that the Multimodal GCN is able to assign higher scores on average to positively associated diseases and genes, as evidenced by the distribution of the inner product of the embeddings in Section 7.1. Finally, we make the observation that the set of diseases for which the Multimodal GCN is unable to make good gene association predictions, is very different to the corresponding set of diseases for Cardigan, in Section 7.2. This hints that an ensemble approach might perform better than either individual approach. Overall, our work provides strong evidence for the utility of multimodal graph convolutional techniques in disease-gene association prediction.

8.1 Future Work

The takeaways observed in the previous section open up some interesting avenues for future work. We list them briefly below:

- In this work, we vary the negative:positive ratio between 1:1, 5:1, and 10:1. As future work, we aim to explore trends for higher negative:positive ratios, like 100:1.
- Here, we perform cross validation using five iterations of repeated random subsampling. It would be worthwhile to perform five-fold cross validation, since it would ensure that every disease/disease-gene association would be in the test set at least once.
- We chose GCN [32] as our convolutional encoder. Since then, other, more complex graph convolutional architectures, have been developed. We leave it to future work to explore the effects of replacing the GCN-based encoder with other techniques, like Graph Attention Networks (GAT) [87].
- We observed an interesting trend in our results, which we were unable to explain. We see that for per-disease cross validation, the performance for Cardigan on the early precision (EP) evaluation measure is higher than the performance of Cardigan on the per-association cross validation. This trend is contrary to what we observed for all the other methods. We leave it to future work to understand why this is the case.
- It would be useful to perform evaluation through temporal holdout. This involves training the model on disease-gene associations collected before a particular point of time, and evaluating it on disease-gene associations uncovered after that point of time.
- It would also be useful to see the top scoring genes in the PPI for every disease, according to the Multimodal GCN, and search the literature for evidence linking the top scoring gene for a disease, with the disease.
- Here, we benchmark our method against supervised learning methods, and a network propagation method. We leave it to future work to benchmark against another graph convolution based method which can make predictions for disease associated genes, or a deep learning based method for disease-gene association prediction.

- We have conjectured that ensemble methods may be useful for this task, since the category of hard to associate diseases (see Section 7.2) is very different for the Multimodal GCN, and Cardigan. We leave it to future work to develop a method which combines network propagation with graph convolution.

Bibliography

- [1] Online mendelian inheritance in man, omim. mckusick-nathans institute of genetic medicine, johns hopkins university (baltimore, md). accessed 07/01/2020. URL <https://omim.org/>.
- [2] Orphanet: an online database of rare diseases and orphan drugs. copyright, inserm 1997. accessed 07/01/2020. URL <http://www.orpha.net>.
- [3] URL <https://www.census.gov/popclock/>.
- [4] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, et al. Gene prioritization through genomic data fusion. *Nature biotechnology*, 24(5):537–544, 2006.
- [5] Mona Alshahrani and Robert Hoehndorf. Semantic disease gene embeddings (smudge): phenotype-based disease gene prioritization without phenotypes. *Bioinformatics*, 34(17):i901–i907, 2018.
- [6] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature reviews genetics*, 12(1):56–68, 2011.
- [7] Anne M Bowcock. Guilt by association. *Nature*, 447(7145):645–646, 2007.
- [8] Liam R Brunham and Michael R Hayden. Hunting human disease genes: lessons from the past, challenges for the future. *Human genetics*, 132(6):603–617, 2013.

- [9] Han G Brunner and Marc A Van Driel. From syndrome families to functional genomics. *Nature Reviews Genetics*, 5(7):545–551, 2004.
- [10] Juan J Cáceres and Alberto Paccanaro. Disease gene prediction for molecularly uncharacterized diseases. *PLoS computational biology*, 15(7):e1007078, 2019.
- [11] Horacio Caniza, Alfonso E Romero, and Alberto Paccanaro. A network medicine approach to quantify distance between hereditary disease modules on the interactome. *Scientific reports*, 5:17658, 2015.
- [12] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *AAAI*, volume 16, pages 1145–1152, 2016.
- [13] Stephen Y Chan and Joseph Loscalzo. The emerging paradigm of network medicine in the study of human disease. *Circulation research*, 111(3):359–374, 2012.
- [14] Helen M Colhoun, Paul M McKeigue, and George Davey Smith. Problems of reporting genetic associations with complex outcomes. *The Lancet*, 361(9360):865–872, 2003.
- [15] Peter Donnelly. Progress and challenges in genome-wide association studies in humans. *Nature*, 456(7223):728–731, 2008.
- [16] Igor Feldman, Andrey Rzhetsky, and Dennis Vitkup. Network properties of genes harboring inherited disease mutations. *Proceedings of the National Academy of Sciences*, 105(11):4323–4328, 2008.
- [17] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. pages 6530–6539, 2017.
- [18] TKB Gandhi, Jun Zhong, Suresh Mathivanan, L Karthick, KN Chandrika, S Sujatha Mohan, Salil Sharma, Stefan Pinkert, Shilpa Nagaraju, Balamurugan Periaswamy,

- et al. Analysis of the human protein interactome and comparison with yeast, worm and fly interaction datasets. *Nature genetics*, 38(3):285–293, 2006.
- [19] Susan Dina Ghiassian, Jörg Menche, and Albert-László Barabási. A disease module detection (diamond) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLoS Comput Biol*, 11(4): e1004120, 2015.
- [20] Greg Gibson. Rare and common variants: twenty arguments. *Nature Reviews Genetics*, 13(2):135–145, 2012.
- [21] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.
- [22] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 855–864, 2016.
- [23] Mellisa Haendel, Nicole Vasilevsky, Deepak Unni, Cristian Bologa, Nomi Harris, Heidi Rehm, Ada Hamosh, Gareth Baynam, Tudor Groza, Julie McMurry, et al. How many rare diseases are there? 2019.
- [24] Peng Han, Peng Yang, Peilin Zhao, Shuo Shang, Yong Liu, Jiayu Zhou, Xin Gao, and Panos Kalnis. Gcn-mf: Disease-gene association identification by graph convolutional networks and matrix factorization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 705–713, 2019.
- [25] S Jane Henley, Elizabeth M Ward, Susan Scott, Jiemin Ma, Robert N Anderson,

- Albert U Firth, Cheryll C Thomas, Farhad Islami, Hannah K Weir, Denise Riedel Lewis, et al. Annual report to the nation on the status of cancer, part i: National cancer statistics. *Cancer*, 126(10):2225–2249, 2020.
- [26] Matthew Herder. What is the purpose of the orphan drug act? *PLoS medicine*, 14(1):e1002191, 2017.
- [27] Borislav H Hristov, Bernard Chazelle, and Mona Singh. ukin combines new and prior information with guided network propagation to accurately identify disease genes. *Cell Systems*, 10(6):470–479, 2020.
- [28] Yu-an Huang, Pengwei Hu, Keith CC Chan, and Zhu-Hong You. Graph convolution for predicting associations between mirna and drug resistance. *Bioinformatics*, 36(3):851–858, 2020.
- [29] Xingpeng Jiang, Bing Liu, Jiefeng Jiang, Huizhi Zhao, Ming Fan, Jing Zhang, Zhenjie Fan, and Tianzi Jiang. Modularity in the genetic disease-phenotype network. *FEBS letters*, 582(17):2549–2554, 2008.
- [30] Simon Jupp, Tony Burdett, Catherine Leroy, and Helen E Parkinson. A new ontology lookup service at embl-ebi. In *SWAT4LS*, pages 118–119, 2015.
- [31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [32] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [33] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

- [34] Sebastian Köhler, Leigh Carmody, Nicole Vasilevsky, Julius O B Jacobsen, Daniel Danis, Jean-Philippe Gourdine, Michael Gargano, Nomi L Harris, Nicolas Matentzoglou, Julie A McMurry, et al. Expansion of the human phenotype ontology (hpo) knowledge base and resources. *Nucleic acids research*, 47(D1):D1018–D1027, 2019.
- [35] Michael Krauthammer, Charles A Kaufmann, T Conrad Gilliam, and Andrey Rzhetsky. Molecular triangulation: bridging linkage and molecular-network information for identifying candidate genes in alzheimer’s disease. *Proceedings of the National Academy of Sciences*, 101(42):15148–15153, 2004.
- [36] Duc-Hau Le. Machine learning-based approaches for disease gene prediction. *Briefings in functional genomics*, 2020.
- [37] Duc-Hau Le and Vu-Tung Dang. Ontology-based disease similarity network for disease gene prediction. *Vietnam Journal of Computer Science*, 3(3):197–205, 2016.
- [38] Duc-Hau Le and Lan TM Dao. Annotating diseases using human phenotype ontology improves prediction of disease-associated long non-coding rnas. *Journal of molecular biology*, 430(15):2219–2230, 2018.
- [39] Yann Le Cam. A hidden priority: the paradox of rarity (eurordis perspective), 2014.
- [40] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.
- [41] Chelsea E Lee, Kaela S Singleton, Melissa Wallin, and Victor Faundez. Rare genetic diseases: Nature’s experiments on human development. *Isience*, page 101123, 2020.

- [42] Yongjin Li and Jagdish C Patra. Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics*, 26(9):1219–1224, 2010.
- [43] Yu Li, Chao Huang, Lizhong Ding, Zhongxiao Li, Yijie Pan, and Xin Gao. Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. *Methods*, 166:4–21, 2019.
- [44] Yu Li, Hiroyuki Kuwahara, Peng Yang, Le Song, and Xin Gao. Pgcn: Disease gene prioritization by disease and gene embedding through graph convolutional neural networks. *bioRxiv*, page 532226, 2019.
- [45] Zhan-Chao Li, Meng-Hua Huang, Wen-Qian Zhong, Zhi-Qing Liu, Yun Xie, Zong Dai, and Xiao-Yong Zou. Identification of drug–target interaction from interactome network with ‘guilt-by-association’ principle and topology features. *Bioinformatics*, 32(7):1057–1064, 2016.
- [46] Núria López-Bigas and Christos A Ouzounis. Genome-wide identification of genes likely to be involved in human genetic disease. *Nucleic acids research*, 32(10):3108–3114, 2004.
- [47] Ping Luo, Yuanyuan Li, Li-Ping Tian, and Fang-Xiang Wu. Enhancing the prediction of disease–gene associations with multimodal deep learning. *Bioinformatics*, 35(19):3735–3742, 2019.
- [48] Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nature methods*, 15(4):290, 2018.
- [49] Urko M Marigorta, Juan Antonio Rodríguez, Greg Gibson, and Arcadi Navarro.

- Replicability and prediction: lessons and challenges from gwas. *Trends in Genetics*, 34(7):504–517, 2018.
- [50] Aaron J Masino, Elizabeth T Dechene, Matthew C Dulik, Alisha Wilkens, Nancy B Spinner, Ian D Krantz, Jeffrey W Pennington, Peter N Robinson, and Peter S White. Clinical phenotype-based gene prioritization: an initial study using semantic similarity and the human phenotype ontology. *BMC bioinformatics*, 15(1):1–11, 2014.
- [51] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.
- [52] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [53] Richard C Mohs and Nigel H Greig. Drug discovery and development: Role of basic biological research. *Alzheimer’s & Dementia: Translational Research & Clinical Interventions*, 3(4):651–657, 2017.
- [54] A Montserrat Moliner and J Waligóra. The european union policy in the field of rare diseases. *Public Health Genomics*, 16(6):268–277, 2013.
- [55] Richard D Moore. Epidemiology of hiv infection in the united states: implications for linkage to care. *Clinical Infectious Diseases*, 52(suppl_2):S208–S213, 2011.
- [56] Fantine Mordelet and Jean-Philippe Vert. Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC bioinformatics*, 12(1):389, 2011.
- [57] Yves Moreau and Léon-Charles Tranchevent. Computational tools for prioritizing

- candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, 13(8): 523–536, 2012.
- [58] Nagarajan Natarajan and Inderjit S Dhillon. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics*, 30(12):i60–i68, 2014.
- [59] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.
- [60] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 809–816, 2011.
- [61] Timothy P O’Connor and Ronald G Crystal. Genetic medicines: treatment strategies for hereditary disorders. *Nature Reviews Genetics*, 7(4):261–276, 2006.
- [62] Anika Oellrich, Robert Hoehndorf, Georgios V Gkoutos, and Dietrich Rebholz-Schuhmann. Improving disease gene prioritization by comparing the semantic similarity of phenotypes in mice with those of human diseases. *PLoS One*, 7(6):e38937, 2012.
- [63] Kenneth Opap and Nicola Mulder. Recent advances in predicting gene–disease associations. *F1000Research*, 6, 2017.
- [64] Kenneth Opap and Nicola Mulder. Recent advances in predicting gene–disease associations. *F1000Research*, 6, 2017.
- [65] Martin Oti and Han G Brunner. The modular nature of genetic diseases. *Clinical genetics*, 71(1):1–11, 2007.
- [66] Martin Oti, Berend Snel, Martijn A Huynen, and Han G Brunner. Predicting disease genes using protein–protein interactions. *Journal of medical genetics*, 43(8):691–698, 2006.

- [67] Rosario M Piro and Ferdinando Di Cunto. Computational approaches to disease-gene prediction: rationale, classification and successes. *The FEBS journal*, 279(5):678–696, 2012.
- [68] Jennifer E Posey. Genome sequencing and implications for rare disorders. *Orphanet journal of rare diseases*, 14(1):153, 2019.
- [69] Stefan M Pulst. Genetic linkage analysis. *Archives of neurology*, 56(6):667–672, 1999.
- [70] Predrag Radivojac, Kang Peng, Wyatt T Clark, Brandon J Peters, Amrita Mohan, Sean M Boyle, and Sean D Mooney. An integrated approach to inferring gene–disease associations in humans. *Proteins: Structure, Function, and Bioinformatics*, 72(3): 1030–1037, 2008.
- [71] Aditya Rao, VG Saipradeep, Thomas Joseph, Sujatha Kotte, Naveen Sivadasan, and Rajgopal Srinivasan. Phenotype-driven gene prioritization for rare diseases using graph convolution on heterogeneous networks. *BMC medical genomics*, 11(1):57, 2018.
- [72] Zahra Razaghi-Moghadam and Zoran Nikoloski. Supervised learning of gene-regulatory networks based on graph distance profiles of transcriptomics data. *NPJ systems biology and applications*, 6(1):1–8, 2020.
- [73] Trevor Richter, Sandra Nestler-Parr, Robert Babela, Zeba M Khan, Theresa Tesoro, Elizabeth Molsen, and Dyfrig A Hughes. Rare disease terminology and definitions—a systematic global review: report of the ispor rare disease special interest group. *Value in Health*, 18(6):906–914, 2015.
- [74] Neil J Risch. Searching for genetic determinants in the new millennium. *Nature*, 405 (6788):847–856, 2000.

- [75] Peter N Robinson, Sebastian Köhler, Sebastian Bauer, Dominik Seelow, Denise Horn, and Stefan Mundlos. The human phenotype ontology: a tool for annotating and analyzing human hereditary disease. *The American Journal of Human Genetics*, 83(5):610–615, 2008.
- [76] Katharina Schwarze, James Buchanan, Jilles M Fermont, Helene Dreau, Mark W Tilley, John M Taylor, Pavlos Antoniou, Samantha JL Knight, Carme Camps, Melissa M Pentony, et al. The complete costs of genome sequencing: a microcosting study in cancer and rare diseases from a single center in the united kingdom. *Genetics in Medicine*, 22(1):85–94, 2020.
- [77] Feichen Shen, Suyuan Peng, Yadan Fan, Andrew Wen, Sijia Liu, Yanshan Wang, Liwei Wang, and Hongfang Liu. Hpo2vec+: Leveraging heterogeneous knowledge resources to enrich node embeddings for the human phenotype ontology. *Journal of Biomedical Informatics*, 96:103246, 2019.
- [78] Vikash Singh and Pietro Lio. Towards probabilistic generative models harnessing graph neural networks for disease-gene prediction. *arXiv preprint arXiv:1907.05628*, 2019.
- [79] Aaron Smalter, Seak Fei Lei, and Xue-wen Chen. Human disease-gene classification with integrative sequence-based and topological features of protein-protein interaction networks. In *2007 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2007)*, pages 209–216. IEEE, 2007.
- [80] Cynthia L Smith, Carroll-Ann W Goldsmith, and Janan T Eppig. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology*, 6(1):R7, 2005.

- [81] Nikunj Somia and Inder M Verma. Gene therapy: trials and tribulations. *Nature Reviews Genetics*, 1(2):91–99, 2000.
- [82] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613, 2019.
- [83] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318, 2019.
- [84] Marc A Van Driel, Jorn Bruggeman, Gert Vriend, Han G Brunner, and Jack AM Leunissen. A text-mining analysis of the human phenome. *European journal of human genetics*, 14(5):535–542, 2006.
- [85] Liese Vandeborne, Eline van Overbeeke, Marc Doods, Birgit De Beleyr, and Isabelle Huys. Information needs of physicians regarding the diagnosis of rare diseases: a questionnaire-based study in belgium. *Orphanet journal of rare diseases*, 14(1):1–11, 2019.
- [86] Oron Vanunu, Oded Mager, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Computational Biology*, 6(1):e1000641, 2010.
- [87] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

- [88] Inder M Verma, L Naldini, T Kafri, H Miyoshi, M Takahashi, U Blömer, N Somia, L Wang, and FH Gage. Gene therapy: promises, problems and prospects. In *Genes and Resistance to Disease*, pages 147–157. Springer, 2000.
- [89] Stéphanie Nguengang Wakap, Deborah M Lambert, Annie Olry, Charlotte Rodwell, Charlotte Gueydan, Valérie Lanneau, Daniel Murphy, Yann Le Cam, and Ana Rath. Estimating cumulative point prevalence of rare diseases: analysis of the orphanet database. *European Journal of Human Genetics*, 28(2):165–173, 2020.
- [90] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1225–1234, 2016.
- [91] Gavin S Wilkie and Eric C Schirmer. Guilt by association: the nuclear envelope proteome and disease. *Molecular & Cellular Proteomics*, 5(10):1865–1875, 2006.
- [92] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li. Network-based global inference of human disease genes. *Molecular systems biology*, 4(1):189, 2008.
- [93] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [94] MaoQiang Xie, YingJie Xu, YaoGong Zhang, TaeHyun Hwang, and Rui Kuang. Network-based phenome-genome association prediction by bi-random walk. *PLoS one*, 10(5):e0125138, 2015.
- [95] Jianzhen Xu and Yongjin Li. Discovering disease-genes by topological features in human protein–protein interaction network. *Bioinformatics*, 22(22):2800–2805, 2006.

- [96] Jianzhen Xu and Yongjin Li. Discovering disease-genes by topological features in human protein–protein interaction network. *Bioinformatics*, 22(22):2800–2805, 2006.
- [97] Peng Yang, Xiaoli Li, Hon-Nian Chua, Chee-Keong Kwoh, and See-Kiong Ng. Ensemble positive unlabeled learning for disease gene identification. *PloS one*, 9(5): e97079, 2014.
- [98] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.
- [99] Evangelia I Zacharaki. Prediction of protein function using a deep convolutional neural network ensemble. *PeerJ Computer Science*, 3:e124, 2017.
- [100] Xiangxiang Zeng, Ningxiang Ding, Alfonso Rodríguez-Patón, and Quan Zou. Probability-based collaborative filtering model for predicting gene–disease associations. *BMC medical genomics*, 10(5):76, 2017.
- [101] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [102] Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [103] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.