Machine Learning Models for Biomedical Ontology
Integration and Analysis

Dissertation by
Fatima Zohra Smaili

In Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy

King Abdullah University of Science and Technology
Thuwal, Kingdom of Saudi Arabia

September, 2020
EXAMINATION COMMITTEE

The dissertation of Fatima Zohra Smaili is approved by the examination committee

Committee Chairperson: Dr. Xin Gao
Committee Members: Dr. Andrey Rzhetsky, Dr. Robert Hoehndorf, Dr. Stefan Arold
Machine Learning Models for Biomedical Ontology Integration and Analysis

Fatima Zohra Smaili

Biological knowledge is widely represented in the form of ontologies and ontology-based annotations. Biomedical ontologies describe known phenomena in biology using formal axioms, and the annotations associate an entity (e.g. genes, diseases, chemicals, etc.) with a set of biological concepts. In addition to formally structured axioms, ontologies contain meta-data in the form of annotation properties expressed mostly in natural language which provide valuable pieces of information that characterize ontology concepts. The structure and information contained in ontologies and their annotations make them valuable for use in machine learning, data analysis and knowledge extraction tasks.

I develop the first approaches that can exploit all of the information encoded in ontologies, both formal and informal, to learn feature embeddings of biological concepts and biological entities based on their annotations to ontologies. Notably, I develop the first approach to use all the formal content of ontologies in the form of logical axioms and entity annotations to generate feature vectors of biological entities using neural language models. I extend the proposed algorithm by enriching the obtained feature vectors through representing the natural language annotation properties within the ontology meta-data as axioms. Transfer learning is then applied to learn from the biomedical literature and apply on the formal knowledge of ontologies.

To optimize learning that combines the formal content of biomedical ontologies and natural language data such as the literature, I also propose a new approach that
uses self-normalization with a deep Siamese neural network that improves learning from both the formal knowledge within ontologies and textual data.

I validate the proposed algorithms by applying them to the Gene Ontology to generate feature vectors of proteins based on their functions, and to the PhenomeNet ontology to generate features of genes and diseases based on the phenotypes they are associated with. The generated features are then used to train a variety of machine-learning based classifiers to perform different prediction tasks including the prediction of protein interactions, gene-disease associations and the toxicological effects of chemicals. I also use the proposed methods to conduct the first quantitative evaluation of the quality of the axioms and meta-data included in ontologies to prove that including axioms as background improves ontology-based prediction.

The proposed approaches can be applied to a wide range of other bioinformatics research problems including similarity-based prediction and classification of interaction types using supervised learning, or clustering.
ACKNOWLEDGEMENTS

I acknowledge with deep gratitude and appreciation, the inspiration, encouragement and guidance given to me by my supervisor Professor Xin Gao. I am deeply grateful to him for his continuous support since I joined KAUST as a masters student several years ago.

Warm thanks are due to Professor Robert Hoehndorf with whom I have closely collaborated. I am deeply grateful for his time, encouragement, immense knowledge and personal involvement during my thesis.

Special thanks to the committee members, Professor Andrey Rzhetsky, Professor Stefan Arold and Professor Robert Hoehndorf for accepting to review this dissertation.

Finally, I would like to express my deepest gratitude to my mother, father, sister and husband for their love, patience and unconditional support throughout all the phases of my academic career.
# TABLE OF CONTENTS

**Examination Committee Page**  
Examination Committee Page 2

**Copyright**  
Copyright 3

**Abstract**  
Abstract 4

**Acknowledgements**  
Acknowledgements 6

**List of Figures**  
List of Figures 11

**List of Tables**  
List of Tables 13

## 1 Introduction

1.1 Background  
1.1.1 Symbolic vs statistical artificial intelligence  16  
1.1.2 Ontologies  17  
1.1.3 Methods for data analysis using biomedical ontologies  21

1.2 Problem statement  
1.2.1 Functional annotation using biomedical ontologies  25  
1.2.2 Knowledge representation from ontologies  26  
1.2.3 Machine learning on biomedical ontologies  26  
1.2.4 Ontology evaluation  27  
1.2.5 Learning from the biomedical literature and ontologies  27

1.3 Objectives and contributions  28

1.4 Thesis outline  30

## 2 Prediction of protein functions

2.1 Introduction  32

2.2 Materials and Methods  
2.2.1 Data set  36

2.2.2 Enzyme Commission (EC) number prediction  36

2.2.3 Gene Ontology (GO) function prediction  39
2.3 Results

2.3.1 Prediction of EC numbers

2.3.2 Prediction of GO classes

2.3.3 Case study

2.3.4 Experimental validation of TRIM22 dimerization

2.4 Discussion and limitations

2.5 Assessment of protein GO function prediction and knowledge extraction from ontologies

3 Knowledge representation from biomedical ontologies

3.1 Introduction

3.2 Materials and Methods

3.2.1 Ontology and annotation resources

3.2.2 Data sets

3.2.3 Automated reasoning

3.2.4 Representation learning using Word2Vec

3.2.5 Text corpora

3.2.6 Similarity

3.2.7 Supervised learning

3.2.8 Clustering and visualization

3.2.9 Text-mining based prediction method

3.2.10 Evaluation using ROC and AUC

3.3 Onto2Vec: Knowledge representation from the formal content of ontologies

3.3.1 Similarity-based prediction of biological relations

3.3.2 Clustering and visualization

3.3.3 Ontologies as graphs and axioms

3.3.4 Towards "trainable" semantic similarity measures

3.3.5 The informal content of ontologies

3.4 OPA2Vec: Knowledge representation from the formal and informal content of ontologies

3.4.1 Predicting interactions between proteins

3.4.2 Predicting gene–disease associations

3.4.3 Discussion of related work

3.4.4 Performance of OPA2Vec using all ontology meta-data annotation properties
3.4.5 Potential for discovery of novel disease–associated genes

3.4.6 Limitations

3.5 Conclusions

3.5.1 Knowledge representation and learning from the logical axioms of ontologies

3.5.2 Learning from the formal and informal content of ontologies

3.5.3 Ontology evaluation using knowledge representation

4 Task-specific ontology evaluation

4.1 Introduction

4.2 Materials and methods

4.2.1 Ontologies

4.2.2 Evaluation data sets

4.2.3 Analysis algorithms

4.3 Task specific evaluation of ontology content

4.3.1 Evaluation of ontology content for functional protein interactions prediction

4.3.2 Evaluation of ontology content for gene–disease associations prediction

4.4 Discussion

4.5 Conclusions

5 Self-normalizing learning on biomedical ontologies

5.1 Introduction

5.2 Materials and methods

5.2.1 Ontologies

5.2.2 Normalization methods

5.2.3 Embedding methods

5.2.4 Siamese neural network

5.2.5 Data sets

5.3 Ontology-based normalization of natural language

5.4 Prediction using ontology normalization

5.4.1 Protein interaction prediction

5.4.2 Gene–disease association prediction

5.4.3 Chemical–disease association prediction

5.5 Discussion

5.6 Conclusions
# LIST OF FIGURES

1.1 The interface between symbolic and statistical AI ........................................... 18
1.2 Part of the graph representation of the Gene Ontology ............................... 20

2.1 Workflow for sequence motif based function prediction in QAUST............. 44
2.2 Precision-recall curves for EC prediction ......................................................... 46
2.3 Precision-recall curves for GO prediction ......................................................... 49
2.4 Precision-recall curves for GO prediction using different structure methods. ......................................................... 52
2.5 GO prediction performance of QAUST using PPI and motifs only .............. 53
2.6 A study case for protein function prediction using QAUST: .................. 55
2.7 Experimental validation of homodimerization function of TRIM22 ............. 58

3.1 Architecture of the neural network used for the classification ...................... 73
3.2 Onto2Vec Workflow ....................................................................................... 78
3.3 ROC curves for PPI prediction for the unsupervised learning methods ....... 81
3.4 ROC curves for PPI prediction for the supervised learning methods ......... 83
3.5 t-SNE visualization of enzyme vectors ............................................................ 85
3.6 The detailed workflow of the feature vector generation pipeline of ............ 93
3.7 Workflow for protein–protein interaction (PPI) prediction using OPA2Vec 94
3.8 AUC values of different methods for PPI prediction for yeast and human 95
3.9 ROC curves for comparing data properties ..................................................... 97
3.10 ROC curves for each prediction method for PPI prediction accuracy ........ 97
3.11 AUC values for gene–disease association prediction .................................... 101
3.12 ROC curves of each prediction method for gene–disease prediction for human and mouse ......................................................... 102
3.13 Performance of OPA2Vec using all vs some annotation properties for PPI prediction ......................................................... 105
3.14 Performance of OPA2Vec using all vs some annotation properties for gene–disease association prediction. 105
4.1 t-SNE-based illustration of the vector embeddings of the classes of each ontology in GO-plus 121
4.2 ROC curves for PPI prediction using GO and GO-Plus using Onto2Vec 123
4.3 ROC curves for PPI prediction using GO and GO-Plus using Node2Vec 123
4.4 ROC curves for gene–disease prediction comparing PhenomeNET with GO (PhenomeNET + GO) to PhenomeNET with GO-Plus 131
4.5 ROC curves for gene–disease prediction comparing PhenomeNET with GO with the meta-data, to PhenomeNET with GO-Plus with meta-data 131
4.6 ROC curves for gene–disease prediction using Node2Vec comparing PhenomeNET with GO, to PhenomeNET with GO-Plus 132
5.1 Architecture of the deep Siamese neural network 142
5.2 An example of the normalization within a class description in the Gene Ontology 145
5.3 Workflow showing the different steps of the proposed normalization-based learning method 147
5.4 ROC curves for PPI prediction 149
5.5 ROC curves for gene–disease association prediction 150
5.6 Overlapping of positive gene–disease associations between BeFree, Annotated OPA2Vec, OPA2Vec and the MGI database (ground truth) 151
5.7 ROC curves for chemical–disease association prediction 153
5.8 Overlapping positive chemical–disease associations between DigChem, Annotated OPA2Vec, OPA2Vec and the CTD database (ground truth) 154
5.9 Visualization of the t-SNE reduced embeddings of 1,000 chemicals colored by their biological role 155
5.10 ROC curves for predicting toxic and therapeutic associations between chemicals and diseases in the CTD data set 156
## LIST OF TABLES

2.1 Accuracy values of EC prediction for QAUST as well as five other methods. .......................... 47

2.2 \( F_{\text{max}} \) values of each branch of GO for QAUST as well as prediction methods. ........ 50

2.3 \( S_{\text{min}} \) (minimum semantic distance) values of each branch of GO for QAUST as well as other prediction methods. .......................... 50

2.4 P-values from the Mann-Whitney U ................. 50

3.1 Parameters used for training the Word2Vec model. ................. 69

3.2 Spearman correlation coefficient values ........................................ 78

3.3 AUC values of ROC curves for PPI prediction ........................................ 79

3.4 Spearman correlation coefficients for PPI prediction ........................................ 83

3.5 AUC values of the ROC curves for PPI interaction type prediction in human. ......... 84

3.6 AUC values of the ROC curves for PPI interaction type prediction in yeast. .......... 84

3.7 Contribution of each annotation property in GO ontology to PPI prediction. ......................... 98

3.8 Area under PR curve (AUPR) for PPI prediction for human and yeast. ......... 99

3.9 AUC values of PPI prediction for human and yeast based on experimental annotations only. ........................................ 99

3.10 Area under PR curve (AUPR) for gene–disease association prediction for human and mouse. ........................................ 102

3.11 Recall values using OPA2Vec for prediction interacting proteins ........................................ 103

4.1 Number of inter-ontology axioms (with an example) in GO-Plus corresponding to each external ontology. ........................................ 117

4.2 AUC values of ROC curves for PPI prediction for GO-Plus and GO. 122

4.3 AUC values of the ROC curves for PPI prediction showing the contribution of the GO-Plus axioms ........................................ 125
4.4 AUC values of the ROC curves for PPI prediction for each external ontology in GO-Plus .................................................. 127
4.5 AUC values of the ROC curves for PPI prediction for different external ontologies with meta-data in GO-Plus using OPA2Vec and OPA2Vec-NN .................................................. 128
4.6 AUC values of ROC curves for gene–disease prediction .................................................. 130
5.1 AUC values of ROC curves for PPI prediction .................................................. 148
5.2 AUC values for gene–disease association prediction .................................................. 149
5.3 AUC values of gene–disease association prediction on the intersection of our data set and the BeFree data set comparing the performance of BeFree to our methods .................................................. 152
5.4 Gene–disease prediction results for orphan diseases .................................................. 158
5.5 AUC values of ROC curves for chemical–disease association prediction .......................... 159
5.6 Performance comparison between the normalization based method and DigChem on chemical–disease association prediction .................................................. 159
5.7 AUC values of ROC curves for predicting therapeutic and toxic associations between chemicals and diseases in the CTD data set .................................................. 159
Chapter 1

Introduction

The past few years have seen a significant increase in the size and complexity of data in the life sciences. Biological knowledge in particular is widely spread across a large number of resources and in different formats. These resources often capture complementary aspects of biological phenomena which increases the challenge of efficient data integration and analysis. Over the years, substantial efforts have been deployed to represent this knowledge in a formal and structured way using biomedical ontologies [1].

Ontologies are now employed at a wide scale in the biomedical field as a result of the main features they provide: classes and relations defined using a unique identifier, a standard domain vocabulary that provides a set of labels to define what a class represents, logical axioms in formal language that enable computational access and exploitation and finally natural language definitions and descriptions (meta-data) that help convey the exact meaning of a class [2].

Ontologies are increasingly being used to annotate biomedical entities, such as genes, diseases and chemicals to classify their biological functions and associations with one or more classes. An ontology-based annotation is usually combined with meta-data about the evidence used, the creator, etc. The Gene Ontology (GO) [3] as an example, is widely used to annotate genes and gene products with their corresponding functions and cellular locations. However, providing reliable GO annotations to proteins computationally is still an open research problem. Knowledge extraction from ontologies is also an open challenge given the lack of methods that can en-
code all of the information within ontologies in ways that are processable by modern machine learning models.

In my thesis, I address both the biological challenge of reliable ontology-based annotation, namely the prediction of GO-based annotations of proteins, and the computational challenge of knowledge representation from ontologies for machine learning models.

In this introductory this chapter, I first expose the background of my research before discussing the research problems I address, and summarizing the key contributions of my work.

1.1 Background

1.1.1 Symbolic vs statistical artificial intelligence

Symbolic (or classical) Artificial Intelligence (AI) was the dominant branch of AI for much of the 20th century [4]. It is based on the philosophical perception of explicitly representing the human knowledge as a set of symbols and rules that manipulate these symbols. Formal propositions are used to assert relations that hold between certain objects. Symbolic systems are then used to infer knowledge by carrying out a series of logic-based reasoning steps [5]. AI systems based on Symbolic AI have had impressive results as was the case with the highly domain-specific Experts Systems that can be thought of as computerised consulting services within very deep but narrow domain of applications [6].

While symbolic AI systems can be successful on well-defined problems, they are generally too constrained to be able to deal with the noise, uncertainty and exceptions of the real world [7]. Nowadays, statistical (modern) AI and specifically machine learning with deep neural networks is in the ascendant to dominate the field of AI. Machine learning models are highly successful in exploiting data to infer knowledge with little need for human experts intervention and are highly noise tolerant. Two
key limitations, however, can affect the performance of these models: (1) their poor generalization sets them for failure when exposed to data outside the distribution they were trained on; (2) some of these models lack interpretability, as is the case of neural networks that are typically thought of as "black boxes" with computations that do not correspond to comprehensible human reasoning steps.

My research lies in the interface between symbolic AI and machine learning. I specifically aim to make the symbolic representations of classical AI processable to machine learning methods that are based on gradient descent. This is based on the assumption that inferring knowledge can be even more efficient if properly done by combining machine learning and symbolic AI in complementary ways (Figure 1.1). I aim to develop models that can preserve the predictive performance of machine learning models using data, while also encoding for the experts knowledge in biomedicine that is available within symbolic systems such as ontologies.

1.1.2 Ontologies

In the context of symbolic AI, an ontology is defined as an "explicit specification of a conceptualization" in a given domain [8]. Ontologies are specifically used to formally describe concepts and relations in a domain using formal axioms that constrain the possible interpretations of the defined concepts [8]. Within an ontology, a class is defined as a partitioning to which some assertion might apply, and an instance is defined as a member of a class (subclass or an individual) [9]. An ontology is expected to provide four key features within the domain where it is defined: (1) defining standard identifiers for classes and relations that can be used across multiple databases, knowledge bases, etc; (2) defining a vocabulary using natural language labels for classes and relation in a domain; (3) a rich meta-data in natural language that provides comprehensive definitions, descriptions and synonyms for classes in the ontology; (4) formal axioms that define relations between the classes in the ontology and that con-
Figure 1.1: The interface between symbolic and statistical AI. My research is mostly in the direction that uses symbolic AI, namely ontologies, as input for statistical AI, namely machine learning models, to produce knowledge.
strain the interpretation of the classes [2]. These features make ontologies widely used
techniques for knowledge representation especially in domains with massive semantic heterogeneity [10]. Using ontologies for knowledge representation as facts has also the benefit of helping make inferences about a domain of study.

Biomedicine has been witnessing a massive increase in data and knowledge in the past few years with the rise of high-throughput sequencing, single cell data, biomedical imaging and text. With this increasing amount of heterogeneous data produced in biology comes a pressing need for methods for the unification, integration and retrieval of this data. To overcome these challenges, biomedical ontologies are now widely used for efficient integration and analysis of data in the biomedical field [2].

The Gene Ontology (GO) [3], for instance, is arguably the most widely used ontology in biomedicine. It has been developed to provide formal and unified descriptions of the molecular functions of genes and gene products, the biological processes where they are involved and the cellular components where they are located [11]. The Gene Ontology has been successful at providing functional annotations for genes and gene products in several organisms. As more and more biologists have adopted the terminology provided by GO, sharing and integrating data across different resources became more efficient [10]. Part of the graphical representation of the Gene Ontology is shown in Figure 1.2.

**Ontologies in OWL**

Biomedical ontologies are mostly formalized using the Web Ontology Language (OWL) [12], a language based on Description Logic, a decidable fragment of first order predicate logic. Description Logics enable a formal, machine-readable description of the types of entities within a domain and the relations in which they stand [13]. OWL comes with explicit semantics that define how statements made in OWL constrain the world in which these statements are interpreted [14].
Figure 1.2: Part of the graph representation of the Gene Ontology [3]. The nodes represent the classes with their identifiers, labels and definitions. The edges represent the subsumption (is-a) relation between classes.
Graph representation of ontologies

Although ontologies in OWL are primarily sets of axioms, many ontology-based analysis methods, including machine learning methods and semantic similarity measures, rely on generating graph representations from the axioms in an ontology [14]. Graph representations of ontologies can be derived dynamically from the formal axioms. There are several ways in which axioms can be used to generate a graph structure.

An important ontology for generating graphs from biomedical ontologies is the OBO Relation Ontology [15] which provides a set of axiom patterns that must hold true for two classes if an edge between them should be created [16]. An axiom pattern is an axiom with variables for classes or individuals; $X \sqsubseteq Y$ is an axiom pattern in which $X$ and $Y$ are variables and if this statement is true for two classes $X$ and $Y$, an edge with label is-a should be created between them: $X \xrightarrow{\text{is-a}} Y$. Axioms can also express disjointness between two classes such as $X \sqcap Y \sqsubseteq \bot$ based on which a disjoint edge can be created ($X \leftrightarrow \text{disjoint}Y$) [14].

The set of axioms used to generate graph representations of ontologies is however only a subset of the much bigger set of logical axioms that can be encoded within an ontology. Therefore, a graph representation does not encode for all the formal content of ontologies. The content of ontologies also goes beyond the formal axioms, to also include rich meta-data that consists primarily of textual labels, descriptions, synonyms, etc. This meta-data is also generally not encoded within the graph representation of ontologies.

1.1.3 Methods for data analysis using biomedical ontologies

As a result of the wide-spread use of ontologies, several tools have been developed to utilize the information encoded in ontologies. Most of these tools fall into one of the following categories: automated reasoners, semantic similarity measures or other methods that primarily use the graphical representation of an ontology.
Automated reasoners

Reasoners are the primary means to perform inference based on ontology axioms semantically [2]. Reasoners use deductive inference to infer knowledge from ontologies in the form of newly inferred logical axioms. They can also be used to answer specific queries mostly about consistency within the ontology. Some of the most widely used automated reasoners are HermiT and Elk: HermiT reasoner [17] is known for its high performance for complex ontologies as it supports the complete OWL 2 DL standard but its worst case complexity is 2-NEXPTIME-complete [18], while Elk reasoner [19] supports only the OWL 2 EL subset of OWL 2 [12] which makes it more efficient in terms of complexity when applied to larger ontologies while losing some possible inferences. While reasoners can in some cases be applied to directly analyze data consistency or infer unknown biological knowledge from the ontology axioms, they need to be used in association with other analytical tools to be able to apply machine learning on ontologies.

Semantic similarity measures

Semantic similarities are widely applied to the similarity-based analysis of ontologies and entities they annotate. A semantic similarity is a measure that can be used to measure the similarity between two or more ontology classes, sets of classes, or entities annotated with sets of ontology classes. Accurate and reliable semantic similarity measures can be particularly useful to validate and analyze studies on biomedical ontologies, such as protein function prediction. Semantic similarity measures can be classified into different types depending on how annotations (or instances) of ontology classes are incorporated or weighted, and the type of information from an ontology that is used to determine the similarity [20, 21]. Among the most widely used semantic similarities is Resnik’s similarity [22], Lin’s similarity [23] and Jiang&Conrath similarity measure [24].
Resnik’s semantic similarity measure is based on the notion of information content (IC) which quantifies the specificity of a given class in the ontology. The information content of a class \( c \) is defined as the negative log likelihood, \(-\log p(c)\), where \( p(c) \) is the probability of encountering an instance or annotation of class \( c \). Given this definition of information content, Resnik similarity is formally defined as:

\[
\text{sim}_{\text{Resnik}}(c_1, c_2) = -\log p(c_{\text{MICA}}),
\]

(1.1)

where \( c_{\text{MICA}} \) is the most informative common ancestor of \( c_1 \) and \( c_2 \) in the ontology hierarchy, defined as the common ancestor of \( c_1 \) and \( c_2 \) with the highest information content value.

Lin’s similarity measure is defined as:

\[
\text{sim}_{\text{Lin}}(c_1, c_2) = \frac{2 \cdot \log p(c_{\text{MICA}})}{\log p(c_1) + \log p(c_2)},
\]

(1.2)

Jiang&Conrath similarity (\( \text{sim}_{\text{J&C}} \)) uses the same components as in Lin’s similarity but with a slightly different formulation:

\[
\text{sim}_{\text{J&C}}(c_1, c_2) = 2 \cdot \log p(c_{\text{MICA}}) - \log p(c_1) - \log p(c_2),
\]

(1.3)

One of the main limitations of semantic similarity measures is that they generally treat ontologies as graphs in which nodes represent classes and edges an axiom involving the connected classes \([20, 21]\). However, not all the axioms in an ontology can naturally be represented as graphs \([15, 25, 26]\). A possible alternative may be to consider all axioms in an ontology when computing semantic similarity; the challenge is to determine how each axiom should contribute to determine similarity beyond merely considering their syntactic similarity. Additionally, semantic similarity measures reduce the information encoded in the ontology to a single point (the similarity
value) which cannot be particularly informative or useful in complex analysis tasks that require more comprehensive and informative features.

Graph-based embedding methods

Learning embeddings is another way to define similarity measures on ontologies. An embedding is a mapping from one mathematical structure to another in such a way that the features of the elements such as their pairwise similarity and distance are somewhat preserved in the new structure. This is mostly useful when the second structure is more suitable for some operations or algorithms such as optimization or gradient descent calculation than the first structure [14]. Consequently, a number of graph-based analysis methods can be applied to learn embeddings from the graph structure of ontologies. Notably, there is a large amount of work on knowledge graph embeddings [27, 28, 29, 30], i.e., a set of feature learning methods applicable to nodes in heterogeneous graphs, such as those defined by Linked Data [31]. These methods can be applied to predict new relations between entities in a knowledge graph, perform similarity-based predictions, reason by analogy, or in clustering [32]. However, while some parts of ontologies, such as their underlying taxonomy or partonomy, can naturally be expressed as graphs in which edges represent well-defined axiom patterns [15, 25], it is challenging to represent the full semantic content of ontologies in such a way [26]. It is possible to materialize the implicit, inferred content of formally represented knowledge bases through automated reasoning, and there is a long history in applying machine learning methods to the deductive closure of a formalized knowledge base [33, 34]. Similar approaches have also been applied to knowledge graphs that contain references to classes in ontologies [35]. However, these approaches are still limited to representing only the axioms that have a materialization in a graph-based format.

Overall, while many tools have been developed for ontology analysis, there is a
lack of methods that can utilize all of the knowledge encoded in ontologies (all axioms and meta-data) and combine it in a meaningful way to be applicable to general data analysis tasks and be used by machine learning tools.

On the other hand, despite the large number of machine learning based analysis of biomedical knowledge (raw sequences, images, literature), there is little effort in utilizing the integrity of the rich content of biomedical ontologies, both formal and informal (meta-data), as input for modern machine learning tools to perform prediction tasks.

1.2 Problem statement

1.2.1 Functional annotation using biomedical ontologies

One of the key objectives of biomedical ontologies is their use for consistent functional annotation of biomedical entities such as genes, diseases, etc [10]. The Gene Ontology, for instance, plays a central role for the functional annotation of genes and gene products. With the increasing availability of entire genome sequences, reliable functional annotation methods for genes with GO classes become of primary importance. While experimental validation is the most reliable way to provide functional annotation to genes and gene products, it is quite expensive in both cost and time. Relying on experimental validation only will therefore lead to a significantly increasing gap between the number of available amino acid sequences and the number of functionally annotated genes and proteins. Providing reliable computational functional annotation to amino acid sequences with unknown structures is a challenge I partially address in my thesis.
1.2.2 Knowledge representation from ontologies

Machine learning approaches are increasingly being used to perform analysis and prediction tasks on biomedical data. However, even with the large amount of data encoded in biomedical ontologies, machine learning algorithms cannot easily be applied on ontologies directly. Extracting information and knowledge from biomedical ontologies in forms that are processable by machine learning models based on gradient descent (or generally, vector operations) is a key intermediate step to allow for the utilization and analysis of this information by machine learning algorithms. Existing methods for knowledge extraction from ontologies such as semantic similarities or graph-based methods can only be applied on the graph representing the taxonomic hierarchies of classes, class definitions and binary relations between classes within the ontology. Ontologies however are not limited to these forms as they also encode other axioms with define more possible constrains on the interpretations of the ontology classes and which are not necessarily included in the graph form of an ontology [8]. In my thesis, I address this challenge by proposing new methods that can extract knowledge from the formal (logical axioms) and informal (meta-data) content of ontologies in ways that make it usable by machine learning without losing any of the information that is otherwise missed in the graph representation of an ontology.

1.2.3 Machine learning on biomedical ontologies

Despite the significance of the number of proposed machine learning models in bioinformatics, most of the proposed approaches are applied on raw or extracted features from biomedical data such as sequences, text or images. With the exception of approaches designed for graph analysis, there is a limited number of machine learning models that can process all of the information included in biomedical ontologies for prediction tasks. Through several biomedical applications in this thesis, I show how machine learning models can be applied on ontologies while also proposing new mod-
els that can handle some of challenges specific to ontologies such as bias in ontology based annotations (e.g. bias in gene disease annotations).

1.2.4 Ontology evaluation

As ontologies are becoming more popular in biomedicine, significant effort is being deployed to constantly improve and enrich the content of ontologies. These improvements consist not only of efforts to improve individual ontologies by including additional classes and axioms and by enriching the meta-data, but also of attempts to create axioms that link classes from different ontologies (e.g. axioms that link gene functions in GO to chemicals in the ontology of Chemical Entities of Biological Interest (ChEBI)) \cite{36, 37, 38}. Despite this significant effort, there are currently no methods that can quantitatively assess the improvement that is added through these efforts. In my thesis, I tackle this challenge by proposing a workflow that can be used to quantitatively evaluate the formal axioms and the meta-data encoded within an ontology as well as quantitatively assess the value of the links and relations created between different ontologies.

1.2.5 Learning from the biomedical literature and ontologies

Every biomedical ontology contains as a form of meta-data, textual labels, descriptions and synonyms associated with the classes in an ontology and provided by experts to convey the precise meaning of the class \cite{2}. This meta-data is rich in technical biomedical words that are also widely used in the biomedical literature. Learning from both the ontology meta-data and the literature in complementary ways has therefore the potential to improve analysis using ontologies but can be counter-productive if the meta-data and the literature are not well aligned with the application of interest. In this work, I address this challenge by proposing a new application-specific learning method that efficiently extracts knowledge from both ontologies and the literature.
1.3 Objectives and contributions

In summary, during my PhD, I developed the first approaches to learn feature representations of biomedical entities from the formal and informal content of biomedical ontologies. The obtained representations can be used as features for machine learning algorithms and can be applied to a wide range of applications.

A key objective of biomedical ontologies is to provide functional annotation for biomedical entities. In my work, I also propose a new computational method to reliably annotate protein sequences with unknown tertiary structures. A key question that arises here however, is related to the assessment of the prediction performance. The precision of this assessment depends on using methods that can accurately evaluate the similarity between GO classes. This is a sub-problem of the much broader research question of knowledge extraction from ontologies.

While there are many methods for knowledge extraction from ontologies, notably a large number of semantic similarity measures [20, 22, 24, 23], there are no methods that can encode for all of the knowledge within ontologies (all axioms and meta-data) as vector representations. Representing the knowledge within ontologies as numerical vectors is, in fact, necessary to allow for the use of ontologies in modern machine-learning models that use gradient descent such as neural networks. In what makes the key contribution of my doctoral work, I propose the first approach, Onto2Vec [39] to learn from all the available logical axioms and all the known ontology annotations and produce real-valued vector representations of both the ontology classes and the annotated entities. I then further extend the approach to OPA2Vec [40] a method that can learn from both the formal (axioms) and informal (meta-data) axioms of the ontology by using the biomedical literature as background. I evaluate my knowledge extraction method by applying them to produce protein feature vectors (based on the Gene Ontology) for protein interaction prediction, and gene and disease feature vectors (based on the PhenomeNet ontology [41]) for gene–disease association
The content of biomedical ontologies is constantly being updated and improved by experts [30]. There are however no methods that can quantitatively show the contribution of the added axioms and meta-data to improve prediction using biomedical ontologies. In my thesis, I perform the first task-specific quantitative assessment of the logical axioms and natural language of several biomedical ontologies towards prediction tasks.

The results from the conducted ontology evaluation show that using axioms as background knowledge contributes indeed to the predictive performance of ontologies. However, the findings also show that combining formal axioms with natural language data does not always improve ontology-based prediction. This is especially the case when the natural language content of the ontology is not well aligned with the analysis task of interest. In the last part of my thesis, I propose a new method that can solve this limitation by using ontology-based normalization to optimize machine learning that uses both formal ontologies and textual data. To further improve the predictive performance of this learning method, I propose a deep Siamese neural network that I test on several biomedical prediction tasks including gene–disease association prediction and the prediction of toxicological effects of chemicals.

The main contributions of this doctoral work can be summarized in the following points:

- A new method to reliably predict functional Gene Ontology annotations for protein sequences with unresolved structures.

- The first approach, Onto2Vec, to learn vector representation of biomedical ontology classes and biomedical entities from all the formal content (logical axioms) of ontologies.

- The first approach, OPA2Vec, to combine the formal (axioms) and informal...
(meta-data) content of ontologies to produce ontology-based representation learning of biomedical concepts and entities.

- A new approach to optimize learning from both formal biomedical knowledge from ontologies and the biomedical literature in machine learning algorithms using ontology-based normalization and a deep Siamese neural network.

- The first quantitative evaluation to show that axioms as background improve the predictive performance of biomedical ontologies for specific tasks.

- Proposing machine learning algorithms that use ontology-based features for different prediction applications (protein interaction prediction, gene–disease association prediction and chemical–disease association prediction) including deep multi-layer perceptron (MLP) networks and deep Siamese neural networks.

1.4 Thesis outline

The rest of this thesis dissertation is organized as follow: In Chapter 2 I propose a new protein function prediction method that can provide reliable ontology-based functional annotation to proteins. In Chapter 3 I propose two new methods for knowledge extraction from biomedical ontologies, Onto2Vec, a new approach to learn joint vector-based representation of biological entities and their ontology-based annotations and its application on the Gene Ontology (GO) and OPA2Vec, that extends the previous approach into combining both the formal and informal content of ontologies to learn feature representations while using the literature as background. In Chapter 4 I introduce a new workflow that uses the proposed approaches to evaluate the quality of ontology content using the Gene Ontology (GO) and PhenomeNet ontology as case studies. In Chapter 5 I propose a novel normalization-based learning method that optimizes the integration of natural language data with formal
knowledge from ontologies. In the last chapter, I conclude the work done in this thesis and discuss possible future directions.
Chapter 2

Prediction of protein functions

2.1 Introduction

One of the key objectives of biomedical ontologies is to provide annotations for biomedical entities [2]. The Gene Ontology (GO) [3] in particular has been used to provide millions of functional annotations to genes and gene products. In this chapter, I propose a new method that can computationally functionally annotate proteins with unknown 3D structure and with no close annotated homologs.

As of today, over 90 million protein sequences are available in the UniProtKB/TrEMBL database [42]. However, this increase in the number of known protein sequences does not reflect a parallel increase in our biological knowledge, as less than 1% of these sequences have a manually annotated function [43]. On the other hand, the functional annotation of these sequences is not only an essential step for the understanding of physiological processes and biological systems in living entities, but also one of the highly challenging tasks in biology, which is why there is an increasing need to provide reliable, automated protein function annotation. Significant efforts have been made to identify evolutionarily related proteins and automatically transfer functional annotations between homologous protein pairs [44, 45, 46, 47]. To make such sequence similarity based functional transfer possible, powerful sequence-alignment methodologies have been developed. In particular, algorithms like BLAST/PSI-BLAST [44] and hidden Markov model (HMM) based techniques [45, 46, 47] have been frequently used to transfer functional annotations between homologous proteins.
The underlying assumption of these sequence-based methods is that evolutionarily related proteins may inherit the function of a shared common ancestor. However, there are numerous cases in which proteins with high sequence similarity have distinct functions [48, 49]. To partially address the problem, several methods have been developed to predict function using annotated conserved sequence motifs that are responsible for the functional aspect of the protein. These methods typically construct the sequence motifs from multiple sequence alignment of proteins belonging to the same protein family with known function [50, 51, 52]. They, however, have two major limitations. First, high-quality sequence alignment is typically required for motif construction, which is not trivial to obtain especially when the sequence homology is low. Second, the accuracy is limited by the quality of functional annotation of motifs. To overcome these limitations, I propose in this part of my work to use a protein-specific “functionally discriminative motif” constructed from sequence fragments excised from the template sequence. From another perspective, the 3-dimensional (3D) structure of a protein sequence is believed to be more involved in its biological function [53, 54] since structures are more conserved than sequences are. The 3D structure of a protein can therefore provide additional information for function transfer, especially when the sequence similarity between related proteins is too low for sequence homolog detection [55, 56]. However, the relationship between the protein function and its structure is not straightforward, as in some cases, similar structures perform the same function while in many cases similar folds perform different functions [57, 58]. Therefore, many prediction methods have been relying on local structure similarity search methods rather than global similarity search to identify functionally homologous proteins [59, 60, 61]. Most of these approaches scan the query protein against a library of known conserved spatial motifs or known active sites (e.g. binding sites) with known function [62]. Local similarity search methods have been proven to be quite accurate in detecting functional similarity between proteins of different folds, but they
also have a high probability of producing false positive matches \[63\]. One possible solution is to combine global and local structure alignment to overcome the promiscuity of global structure comparison and low specificity of local structure matching \[64, 65\], which I implement in this part of my thesis. A number of function prediction methods are based on the information extracted from protein-protein interaction (PPI) networks \[66, 67\]. The assumption in this case is that proteins that physically interact with each other frequently appear at the same sub-cellular location and are part of the same biological process \[68\]. However, it is not always the case that proteins which interact with each other share the same molecular function (e.g. PD1 and PD-L1), which is why PPI information is not always sufficient to predict very specific functions \[69\]. Finally, recently there is an emergence of methods which combine multiple sources of information (PPI, domains, sequence alignments, etc.) using advanced machine learning algorithms to perform function prediction. These methods have shown to improve the prediction performance over methods that use only one type of information \[70, 71, 72, 73, 74, 75, 76, 77\]. In this part of my thesis, I propose a new protein function prediction method, Quantitative Annotation of Unknown Struc- ture (QAUST), which combines the global and local structure similarity search with protein-protein interaction networks and functional sequence motif detection. Our approach follows a sequence-to-structure-to-function workflow. Starting from the protein amino acid sequence, I first generate structure predictions by the Iterative Threading ASSEMBly Refinement method (I-TASSER) \[78\]. The predicted structure is then used to identify the proteins with similar functions based on a combination of global and local structure similarity search method that follows the same pipeline used in COFACTOR \[65, 79\]. Protein-protein interaction information is meanwhile extracted from the STRING database \[80\]. And finally, I extract functionally discriminative sequence motifs as our third main prediction feature. The confidence scores obtained from these three features are combined in a consensus
function to obtain our final confidence score. When validating the proposed method, I make sure to exclude templates having a sequence identity > 30% with the query proteins from the template libraries to eliminate any structure or function homologs to the query. Since the terminology of a “protein function” might be ambiguous, I would like to clarify that the definitions of function followed in this work is Enzyme Commission (EC) numbers [81] and Gene Ontology (GO) classes [82]. EC numbers are used to categorize enzymes into hierarchical families using a numerical classification. Specifically, the EC number (which is composed of four numbers separated by periods i.e. A.B.C.D) refers to the reaction catalyzed by a specific enzyme. On the other hand, the GO classes are a set of controlled vocabulary to formally describe proteins and RNAs based upon their functions. Three aspects of functions, Biological Process (BP), Cellular Component (CC) and Molecular Function (MF), are defined in this database. While GO contains much more information than what is included in its graph representation as I have discussed in the previous chapter, subsets of GO can be represented by a structured directed acyclic graph (DAG) for simplicity, where nodes represent GO classes which describe gene product functions, while the edges represent the relationships ("is_a" or "part_of") between the GO classes. In GO’s functional hierarchy, the more general functions are on the top of the graph while more specific classes are usually present further down the graph. Our prediction results are compared to the following programs: COFACTOR [65], a global and local structure similarity-based method, LOMETS [83], a meta-threading algorithm, HH-search [46], an HMM-based method that is widely used to detect protein homologs, BLAST [44], which transfers annotations based on sequence similarity, naïve baseline which predicts GO classes based on their annotation frequency, as well as two highly-ranked methods from the CAFA assessment [84], GoFDR [73] and INGA [85].
2.2 Materials and Methods

2.2.1 Data set

To evaluate QAUST for EC prediction, my collaborators and I used the benchmark data set of COFACTOR [65] as our testing data set. This data set consists of 318 enzymes with unique EC numbers (first three digits) covering all 6 enzyme classes. Similarly, all sequences in the template libraries with a sequence identity > 30% with the query enzymes are excluded from the template libraries. I evaluate QAUST for GO prediction on a data set of 500 randomly chosen non-redundant proteins from the CAFA 2 targets (https://biofunctionprediction.org/cafa/) annotated with at least one GO class. To eliminate any structure or function homologs to the query, templates having a sequence identity > 30% with the query proteins are excluded from the template libraries both in the I-TASSER threading library and our function prediction template libraries.

2.2.2 Enzyme Commission (EC) number prediction

Global similarity search

The first step of our protein function prediction is the generation of the predicted 3D model of the query protein using I-TASSER [78]. The predicted model of the query protein obtained from I-TASSER is then scanned against a non-redundant (pairwise sequence identity no more than 90%) structure template library of 2,385 enzymes with at least the first three digits of EC number annotated by the Catalytic Site Atlas (CSA) database [86]. This library scanning detects homologous structure templates to the query proteins using two types of structure similarity search programs: global similarity search and local similarity search. Templates with a similar global structure to the predicted structure of the query protein are detected from the template library using TM-align [87]. Another important consideration when searching for templates
with similar global folds to the query protein is the quality of the structural models. Appraising the accuracy of the structure modeling in the scoring scheme helps to reduce the number of false positive predictions. In this particular case, the quality of the predicted I-TASSER model generated in the previous step is evaluated using Cscore [78].

**Local similarity search**

The local structural search approach consists of three steps. The first of which is the structural match of the specific catalytic/active residue pairs. For a given pair of query and template proteins, we first scan the known catalytic/active residues of the template through the query sequence. The query’s residues whose amino acid types are the same as the amino acid types of the template’s catalytic/active residues are marked as potential active sites in the query. The structures of all combined sets of marked residues in the query are extracted from the predicted model and used as candidate active sites. The structure of the candidate site is superimposed on the known catalytic/active residues in the template. To make the structure superimposition more reliable, for each residue i, the coordinates of $C\alpha$ atoms and side-chain centers of mass of the two neighboring residues, i.e. the $i-1$ and $i+1$th residues are also included in the superimposition. The second step is to identify the key local environment residues around the active sites in the query and the template. For this purpose, we superimpose the complete structure of the query and template proteins based on the rotation matrix obtained from the superimposition of the candidate catalytic/active residue structures obtained in previous step. A sphere of radius $r$ is then defined around the geometric center of the template’s local 3D fragments, where $r$ is the maximum distance of the template residues in the local 3D fragment from the geometric center. The sphere represents a local environment or probable active site region, under which the query and template’s chemical and structural similarity are
compared. Because a sphere comprising of a very small number of catalytic/active residues can easily generate false positive hits, when the template’s active site region is small, we set the number of residues inside the sphere to be a minimum of 20 residues. This value is obtained using minimum grid search parameter optimization by evaluating different sphere sizes in the range of [10, 50] residues to select the most accurate value. In the third step, the best alignment of the local active site residues in the spheres between the query and the template is identified using a scoring function similar to TM-align. Starting from the initial superposition of the query and template protein structures, we perform a Needleman-Wunsch dynamic programming to generate the best alignment for the residues in the selected sphere of the template and the query, where the alignment score matrix $S_{ij}$ for aligning the $i$th residue in the query and the $j$th residue in the template is defined as:

$$S_{ij} = \frac{1}{1 + \frac{d_{ij}}{d_0}} + M_{ij}, \quad (2.1)$$

where $d_{ij}$ is the $C\alpha$ distance between residues $i$ and $j$, $d_0$ is the distance cutoff given by obtained from TM-align, $M_{ij}$ is the substitution score between the $i$th and $j$th residues taken from the BLOSUM62 mutation matrix with the value normalized by the diagonal element in the mutation matrix. The gap penalty is set as -1. For a given scoring matrix $S_{ij}$, a new alignment is generated by dynamic programming. A new superposition and scoring matrix are then constructed based on the new alignment to obtain a newer alignment from dynamic programming. This procedure is iteratively repeated until the final alignment is converged. For each alignment, the active site match (AcM) is evaluated using an alignment score defined as:

$$AcM = \frac{1}{N_t} \sum_{i=1}^{N_{als}} \frac{1}{1 + \left(\frac{d_{ii}}{d_0}\right)^2} + \frac{1}{N_t} \sum_{i=1}^{N_{als}} M_{ii}, \quad (2.2)$$

where $N_t$ represents the number of residues in the active site sphere of the tem-
plate, $N_{ali}$ is the number of aligned residue pairs. The maximum AcM score obtained during the heuristic iterations is recorded for each candidate active site. Finally, the set of residues in the candidate active site which has the highest AcM score is selected to evaluate the similarity between the query and the template’s active site. The weights and form the AcM score have been derived based on the predicted structures of 100 randomly chosen training proteins from the template library, which are non-homologous (sequence similarity < 30%) to the test proteins in order to maximize the sensitivity and specificity of the predictions. Scoring function for global and local similarity search. The final score for predicting EC numbers, used to sort the hits from the enzyme library is a combination of the global similarity search score and the AcM score (obtained from the local similarity search) and is defined as:

$$QAUSTEC = C_{norm}[TM + \frac{Cov}{1 + RMSD_{ali}}] + 2 \cdot ID_{ali} \cdot Cov + \frac{AcM}{2},$$

(2.3)

where $Cov$ represents the coverage of the structural alignment, $RMSD_{ali}$ is the RMSD (root-mean-square deviation) between the model and the template structure in the structurally aligned region, and $ID_{ali}$ is the sequence identity between query and template based on the alignment generated by TM-align. The hyperbolic-tangent-like normalization is further used to normalize the raw EC score to be between 0 and 1:

$$QAUSTEC_{norm} = \frac{2}{1 + \exp(-QAUSTEC)} - 1$$

(2.4)

2.2.3 Gene Ontology (GO) function prediction

For the prediction of GO functions, we combine three different predictors. Each one of these predictors generates a confidence score. The three confidence scores obtained are then combined in a consensus function to generate the final prediction score. We use
three predictors: first, the global structure similarity, which uses I-TASSER to predict the 3D structure of the query, and then scans a library of templates to identify those which have a similar global structure to the predicted model. The second predictor is based on PPI information, and the third one is based on extracted functional sequence motifs.

**Global protein structure similarity**

Similar to EC prediction, I-TASSER is also used here to construct the corresponding 3D model to the query sequence. The model obtained is then scanned against a library of templates to identify those which share a similar global structure to the query model ([https://zhanglab.ccmb.med.umich.edu/BioLiP/library.html](https://zhanglab.ccmb.med.umich.edu/BioLiP/library.html)). For the time being, the functionally important residues for most of the proteins in the GO template library are unknown. Therefore, only the global similarity search is taken into consideration when sorting the hits from the GO library. Global similarity search for GO prediction is done in a similar way to global similarity search for EC prediction described in the previous section. The only difference is that to select the best hits for GO prediction, we rank a template using the $Fh−score$ defined as:

$$FH_{score} = C_{norm}(TM_{score} + \frac{1}{1 + RMSD_{ali}} \cdot Cov) + 3 \cdot ID_{ali} \cdot Cov \quad (2.5)$$

Since each single protein can be annotated with multiple GO classes and the global search may result in many close template structures, a query protein can have multiple GO function predictions with high $Fh − scores$. Therefore, the confidence score of each GO class is calculated as follow:

$$P_{structure}(\lambda) = \frac{1}{N} \sum_{i=1}^{N_\lambda} Fh(i), \quad (2.6)$$

where $\lambda$ represents a given GO class, $N_\lambda$ is the number of templates annotated
with the GO class $\lambda$, and $N$ is the total number of templates selected for generating the consensus. When multiple close templates are available, we only consider the templates with an $F_h$ score $> 1$. For those query proteins with less than 10 templates of $F_h$ score $> 1$, the top 10 templates are selected for generating the consensus prediction regardless of the $F_h$ score. Also, given the hierarchical nature of the GO DAG, we consider that when a protein is annotated with a given GO class, all its ancestor GO classes (through “is_a” relation) are automatically implied. Therefore, once a GO class $\lambda$ is scored, we score all its ancestor classes as well. The score of any ancestor GO class $\mu$ of class $\lambda$ is calculated as:

$$P_{\text{structure}}(\mu) = P_{\text{structure}}(\lambda) \cdot (1 + \frac{N_\mu}{N_0})$$

where $N_\mu$ and $N_0$ are the number of leaf nodes under node $\mu$ and the root node, respectively.

### Protein-protein interaction network

We exploit the information provided by the STRING [80] database, which is a library of PPI networks, to extend our prediction set. The query protein sequence is mapped to its corresponding STRING entry by BLAST, with minimum sequence identity cutoff of 90%. Extracting the PPI partners of the query, we calculate the confidence score of STRING for a GO class $\lambda$ ($P_{\text{STRING}}(\lambda)$) as the frequency of the GO class $\lambda$ among the experimentally annotated interaction partners of the query protein:

$$P_{\text{STRING}}(\lambda) = \frac{n_\lambda}{N},$$

where $n_\lambda$ is the number of interaction partners annotated with the GO class $\lambda$ and $N$ is the number of partners associated with class $\lambda$, according to the corresponding UniProt-GOA (http://www.ebi.ac.uk/GOA) entry of this PPI partner. This score
could take any value from 0 to 1.

**Functionally discriminative sequence motifs**

In addition to the structure similarity search and protein-protein interaction features discussed above, we also include sequentially extracted features to predict GO classes since a sequence is a highly valuable source of information that can especially be useful when dealing with proteins for which we cannot construct a good quality 3D structure model or those with no known protein-protein interaction information. Our functionally discriminative motif detection algorithm follows three steps: detection of sequence templates, identification of functionally discriminative motifs given a GO class, and scoring the query protein. Detection of sequence templates for query protein: The sequence homologs of the query sequence are detected by PSI-BLAST from the Uniref90 database [88]. We filter all obtained homologs with sequence identity > 30% to the query. Identification of functionally discriminative motifs given a GO class: We map all the selected sequence homologs of the query to their corresponding GO annotations in the UniProt-GOA database (http://www.ebi.ac.uk/GOA). GO classes assigned with “Inferred from Electronic Annotation” (IEA) or “No biological Data available” (ND) evidence codes are not considered. We also filter out annotations with evidence code IPI (Inferred from Physical Interactions) since we use protein-protein interaction (PPI) information in our features. After filtering these annotations, we are left with the annotations based on evidence codes: EXP, IDA, IMP, IGI, IEP, TAS and IC. For each GO class $\lambda$, we build two sets of sequences from the set of homolog sequences detected in the previous step. These two sets are: the “annotated set”, which is the set of sequence homologs annotated with this specific GO class, and the “not-annotated set”, which is the set of sequence homologs not annotated with this given GO class. For each one of these two sets, we extract the ten most frequent motifs by extracting all unique amino acid motifs of length 45, 48 from
the sequence set using sliding windows. These motifs are ranked in descending order by their occurrences. The top 10 most frequent motifs are the initial "frequent list", while the remaining motifs are in the "waiting list". If, within the "frequent list", a short motif is a substring of another longer motif, the shorter motif is discarded, and the most frequent motif from the "waiting list" is transferred to "frequent list" to ensure that the latter always has 10 motifs. This process is iterated until, in the "frequent list", any motif is not a substring of another motif. The motifs in the "frequent list" are used for matching the query in the next step. Scoring the query protein: For each of the two sets (annotated and not-annotated sets) we check the number of frequent motifs extracted in the previous step that are also present in the query sequence. Then, we calculate the confidence score of the GO class $\lambda$ given the query sequence as follows:

$$P_{MOTIF}(\lambda) = \frac{n_q(\lambda)}{N(\lambda)} \left[1 - \frac{n_q(\lambda^c)}{N(\lambda^c)}\right], \quad (2.9)$$

where $\lambda$ is the given GO class, $N(\lambda)$ and $N(\lambda^c)$ are the number of frequent patterns from the "annotated set" and "not-annotated set", both of which equal to 10. $n(\lambda)$ and $n(\lambda^c)$ are the corresponding number of matched patterns at the query sequence. This score can take any value from 0 to 1. An ideal value of this score would be equal to 1, which happens when all the sequences in the annotated set contain these frequent motifs and none of the sequences in the not-annotated set contains these same motifs. This scoring function has been designed to penalize the prediction in case the query sequence matches a high number of frequent motifs from the not-annotated set. This way, the scoring function accounts for two essential pieces of information: which set has the maximum number of frequent motifs matched in the query, and how significant is the difference between the number of matched motifs from the annotated set to that from the not-annotated set. Figure 2.1 shows a flowchart detailing the three steps of extracting functional sequence motifs.
Figure 2.1: **Workflow for sequence motif based function prediction in QAUST.** The query sequence is searched against UniRef90 database by PSI-BLAST to identify sequence homologs with GO class annotation. For a GO class of interest, $\lambda$, the identified homologs are divided into two sets: the “annotated set” (purple) which contains homologs annotated with $\lambda$, and “not-annotated set” (green) which consists of homologs not associated with $\lambda$. From each of the two sets, frequent motifs, i.e. continuous sequence fragments, are extracted. For illustration purposes, only three five-residue-long motifs from each set are drawn. The GO class $\lambda$ is predicted with confidence score $\frac{n_q(\lambda)}{N(\lambda)} \left[ 1 - \frac{n_q(\lambda^c)}{N(\lambda^c)} \right]$. Here, $N(\lambda)$ and $N(\lambda^c)$ are the total number of extracted frequent motifs for “annotated set” and “not-annotated set”, correspondingly; while $n_q(\lambda)$ and $n_q(\lambda^c)$ are the number of frequent motifs from the “annotated set” and “not-annotated set” that match the query sequence, respectively. In this example, only the motif $CLPFD$ from the “annotated set” matches the query, making the confidence score equals to $\frac{1}{3}[1 - \frac{0}{3}] = \frac{1}{3}$. 
Consensus

To predict GO classes, the three main scores obtained from the three different predictors (the structure search, the PPI network, and the functional motifs) are combined by consensus averaging to calculate the final confidence score $P_{\text{consensus}}(\lambda)$ for a GO class $\lambda$:

$$P_{\text{consensus}} = 1 - \prod_{m \in \{\text{structure}, \text{STRING}, \text{MOTIF}\}} (1 - P_m(\lambda)) \quad (2.10)$$

This equation used to calculate the consensus has been previously used by other methods for protein function prediction [85]. If one or more predictors are not available for a given class (e.g. no interaction partners are known for the given query), only the available predictors are used to obtain the confidence score. Also, since GO uses the true-path rule (i.e. if a protein is associated by a term, it is also implicitly annotated by its ancestors), for every predicted GO class, all its ancestors are considered to be predicted as well since they are more general classes.

2.3 Results

2.3.1 Prediction of EC numbers

We compared the EC prediction performance of our method to five methods: HH-search [46], LOMETS [83], BLAST [44], COFACTOR [65] and DEEPre webserver [89]. We compared the performance of these methods based on precision (positive predictive value) and recall (sensitivity) rates. Figure 2.2 shows the precision-recall graph corresponding to four baseline methods as well as QAUST. Since the DEEPre webserver does not report the confidence score with the annotation, we could not draw the precision-recall curves but compared QAUST to DEEPre based on accuracy. An EC number prediction is considered to be “true” if the first three digits of the EC number from the hit are identical to those of the query protein; otherwise the hit is considered to be “false”. As shown in Figure 2.2, the rate of true positive pre-
Figure 2.2: Precision-recall curves for EC prediction by QAUST, COFACTOR, LOMETS, HHsearch and BLAST.

dictions using the EC-score is much higher than that of HHsearch, LOMETS, BLAST and COFACTOR at most recall rates. QAUST has also an area under precision-recall curve (AUPRC) of 0.712 which is higher than that of COFACTOR (0.643), LOMETS (0.510) and HHsearch (0.4894). Table 2.1 reports the accuracy of QAUST compared to five other methods including DEEPre and COFACTOR. The results show that DEEPre has a slightly higher performance than QAUST in terms of accuracy, which is probably due to the fact that DEEPre is a machine learning method trained on a large number of enzymes with known functions that overlap or contain close homologs to our test data.
Table 2.1: Accuracy values of EC prediction for QAUST as well as five other methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>QAUST</td>
<td>0.709</td>
</tr>
<tr>
<td>Cofactor</td>
<td>0.698</td>
</tr>
<tr>
<td>LOMETS</td>
<td>0.661</td>
</tr>
<tr>
<td>HHsearch</td>
<td>0.607</td>
</tr>
<tr>
<td>BLAST</td>
<td>0.571</td>
</tr>
<tr>
<td>DEEPre</td>
<td>0.714</td>
</tr>
</tbody>
</table>

2.3.2 Prediction of GO classes

To assess the contribution of individual predictors to the GO prediction performance by QAUST, we visualize the precision-recall curve of the structure similarity search alone ($P_{\text{structure}}$), the precision-recall curve of structure similarity search combined with PPI information ($P_{\text{structure}}$ and $P_{\text{STRING}}$), and that of the final QAUST prediction ($P_{\text{structure}}$, $P_{\text{STRING}}$ and $P_{\text{MOTIF}}$). Additionally, we compared the prediction performance on our data set (please see subsection data set under section Methods) to COFACTOR [65], BLAST [44], LOMETS [83], HHsearch [46], INGA [85] webserver, a method that combines BLAST, PPI information and Pfam in one predictor, and GoFDR [73], one of the top function prediction methods at the CAFA assessment [84] which uses a machine learning model as classifier and discriminative residues as the main feature. The performance has been primarily evaluated using precision-recall curves computed at each prediction score threshold. We also used the $F_{max}$ measure as a quantitative measure to evaluate the overall performance of the precision-recall curves. Precision, recall, and $F_{max}$ are defined in the same way as the CAFA evaluation [90]. The $F_{max}$ measure has been computed as the maximum value of the $F_{\text{measure}}$ which is computed at each threshold as $\frac{2 \text{precision} \cdot \text{recall}}{\text{precision + recall}}$. Precision at threshold $t$ is defined as $\frac{|P_x(t) \cap C_x|}{|P_x(t)|}$, while recall is defined as $\frac{|P_x(t) \cap C_x|}{|C_x|}$ where $x$ is a query protein, $P_x(t)$ is the set of predicted classes for $x$ at threshold $t$ and $C_x$ is the set of correct classes that $x$ is experimentally annotated with. Sim-
ilar to the CAFA evaluation \[84\], we also reported the minimum semantic distance \((S_{min})\) as an additional evaluation metric for GO prediction. \(S_{min}\) is defined as 
\[
min_t \sqrt{ru(t)^2 + mi(t)^2},
\]
where \(ru(t)\) is the remaining uncertainty at threshold \(t\) defined as 
\[
\frac{1}{n_e} \sum_{i \in e} \sum_f ic(f)|f \notin P_i(t) \land f \in C_i\] and \(mi(t)\) is the misinformation at threshold \(t\) defined as 
\[
\frac{1}{n_e} \sum_{i \in e} \sum_f ic(f)|f \in P_i(t) \land f \notin C_i\] and \(mi(t)\), where \(n_e\) is the number of proteins in our data set, \(P_i(t)\) is the set of predicted GO class for protein \(i\) at threshold \(t\), \(C_i\) is the set of classes that protein \(i\) is actually annotated with and \(ic(f)\) is the information content of the GO class \(f\). The very general and unspecific GO classes such as “Molecular Function”, “Biological Process”, “Cellular Component”, “Binding” and “Protein Binding” are excluded from the evaluation.

As shown in Figure 2.3, our method combining structure, PPI and functional motif information achieves higher precision than most other methods at most recall points, in particular for MF and BP. For our data set, structure and motif information has been used for all proteins. However, the PPI information from STRING was missing for 74 proteins. In this case, only the structure and motif information is used. The \(F_{max}\) measure values are reported in Table 2.2 while the minimum semantic distance \((S_{min})\) values are reported in Table 2.3. Surprisingly, in CC prediction, naïve baseline which predicts GO classes based on their annotation frequency, achieves higher performance than all other methods including QAUST. In fact, in the CAFA assessment \[84\], naïve baseline also outperformed most of the other methods in predicting CC classes. One possible explanation for why naïve baseline has higher \(F_{max}\) for CC classes prediction is because the most frequently used CC classes in protein annotation are usually part of a small set of very general classes such as “cytoplasm” or “intracellular part”. Since the naïve baseline is solely based on frequency, it increases the chance of predicting a true positive \[84\]. We also reported the p-values obtained from the Mann-Whitney U test to assess the significance of the difference in performance of QAUST compared to all other methods in Table 2.4.
Figure 2.3: Precision-recall curves for GO prediction. GO prediction performance of our method based on different sets of features, and seven other methods for each of the three GO branches.
<table>
<thead>
<tr>
<th>Method</th>
<th>MF</th>
<th>CC</th>
<th>BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure(Cofactor)</td>
<td>0.467</td>
<td>0.402</td>
<td>0.367</td>
</tr>
<tr>
<td>Structure + PPI</td>
<td>0.507</td>
<td>0.453</td>
<td>0.436</td>
</tr>
<tr>
<td>Structure + PPI + Motifs(QAUST)</td>
<td>0.568</td>
<td>0.467</td>
<td>0.448</td>
</tr>
<tr>
<td>NaiveBaseline</td>
<td>0.315</td>
<td>0.492</td>
<td>0.387</td>
</tr>
<tr>
<td>LOMETS</td>
<td>0.396</td>
<td>0.374</td>
<td>0.303</td>
</tr>
<tr>
<td>HHsearch</td>
<td>0.381</td>
<td>0.356</td>
<td>0.347</td>
</tr>
<tr>
<td>INGA</td>
<td>0.501</td>
<td>0.436</td>
<td>0.421</td>
</tr>
<tr>
<td>BLAST</td>
<td>0.347</td>
<td>0.373</td>
<td>0.321</td>
</tr>
<tr>
<td>GoFDR</td>
<td>0.579</td>
<td>0.449</td>
<td>0.431</td>
</tr>
</tbody>
</table>

Table 2.2: \(F_{\text{max}}\) values of each branch of GO for QAUST as well as prediction methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>MF</th>
<th>CC</th>
<th>BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure + PPI + Motifs(QAUST)</td>
<td>7.66</td>
<td>5.41</td>
<td>10.80</td>
</tr>
<tr>
<td>Cofactor</td>
<td>7.51</td>
<td>5.81</td>
<td>11.72</td>
</tr>
<tr>
<td>NaiveBaseline</td>
<td>8.26</td>
<td>5.12</td>
<td>12.09</td>
</tr>
<tr>
<td>LOMETS</td>
<td>8.11</td>
<td>7.23</td>
<td>14.56</td>
</tr>
<tr>
<td>HHsearch</td>
<td>8.33</td>
<td>7.68</td>
<td>14.20</td>
</tr>
<tr>
<td>INGA</td>
<td>7.95</td>
<td>6.74</td>
<td>12.27</td>
</tr>
<tr>
<td>BLAST</td>
<td>8.42</td>
<td>6.34</td>
<td>14.01</td>
</tr>
<tr>
<td>GoFDR</td>
<td>7.32</td>
<td>5.60</td>
<td>11.65</td>
</tr>
</tbody>
</table>

Table 2.3: \(S_{\text{min}}\) (minimum semantic distance) values of each branch of GO for QAUST as well as other prediction methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>MF</th>
<th>CC</th>
<th>BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cofactor</td>
<td>0.024</td>
<td>0.044</td>
<td>0.047</td>
</tr>
<tr>
<td>NaiveBaseline</td>
<td>0.009</td>
<td>0.039</td>
<td>0.029</td>
</tr>
<tr>
<td>LOMETS</td>
<td>0.012</td>
<td>0.018</td>
<td>0.011</td>
</tr>
<tr>
<td>HHsearch</td>
<td>0.019</td>
<td>0.012</td>
<td>0.016</td>
</tr>
<tr>
<td>INGA</td>
<td>0.036</td>
<td>0.024</td>
<td>0.041</td>
</tr>
<tr>
<td>BLAST</td>
<td>0.014</td>
<td>0.014</td>
<td>0.011</td>
</tr>
<tr>
<td>GoFDR</td>
<td>0.033</td>
<td>0.041</td>
<td>0.047</td>
</tr>
</tbody>
</table>

Table 2.4: P-values from the Mann-Whitney U test to assess improvement/decrease of QAUST performance in GO prediction compared to other methods.

As a further analysis, to investigate if the performance of our method is solely due to the power of the I-TASSER structure prediction we used, we replaced the I-TASSER structure prediction component of our method by HHsearch and LOMETS.
structure prediction, respectively. Our results show that no matter which structure prediction method is used, our scoring function, \( P_{structure} \), can significantly improve the performance on predicting the GO classes. Meanwhile, among the three structure prediction methods, I-TASSER with \( P_{Structure} \) consistently performs the best over all three GO hierarchy branches of MF, BP and CC, whereas LOMETS with \( P_{Structure} \) has the second best performance on MF and CC, and HHsearch with \( P_{Structure} \) is the second best on predicting BP classes (Figure 2.4). Additionally, we have evaluated the performance of our method when only PPI and motif information are used without including any structure-based information. The results show that the function prediction performance drops when structure features are not used (Figure 2.5).

**How do protein-protein interaction information and functional sequence motifs improve the prediction?**

Protein-protein interaction information extracted from STRING is an important feature used in our prediction. In Figure 2.3, we show how protein-protein interaction information alone improves the performance achieved by the structure similarity search (orange dash lines versus magenta dash lines). The precision-recall curves in Figure 2.3 show that the contribution of protein-protein information from STRING is very significant for CC and BP classes, especially for large recall rates. Moreover, the precision-recall curves confirm our initial hypothesis on the utility of PPI information for function annotation. As shown in the figure, while there is some improvement in predicting MF classes, this improvement is not substantial. The reason PPI is not particularly helpful in MF class prediction is most probably because proteins that interact with each other do not necessarily share the same specific molecular function, even when they are part of the same biological process. In addition to the structure similarity search and the protein-protein interaction features, the results show that the functional motifs extracted improve the performance of the prediction significantly. In addition to comparing the performance of our method to BLAST,
Figure 2.4: **Precision-recall curves for GO prediction using different structure methods.** Precision-recall curves for GO prediction by QAUST’s structure similarity based pipeline (I-TASSER + $P_{\text{structure}}$), alternative implementations using low resolution homology models (HHsearch + $P_{\text{structure}}$ and LOMETS + $P_{\text{structure}}$) and baseline algorithms (HHsearch and LOMETS).
Figure 2.5: GO prediction performance of QAUST compared to the prediction performance when using PPI and motif features only for each one of the three GO branches.
LOMETS, HHsearch, and COFACTOR, we also compared it to INGA and GoFDR, two top methods from CAFA in particular for MF and BP classes prediction, and to naïve baseline which is one of the performance references used in CAFA.

### 2.3.3 Case study

To better illustrate the performance of QAUST and the contribution of each component to the prediction, we used as an example Bacteriophage T4 gene 59 helicase assembly protein (P13342) (the cyan structure in Figure 2.6(A)), which is a DNA binding protein required mainly for DNA replication in the late stage of T4 infection [91]. Figure 2.6(B) shows the set of BP classes associated with this protein. In this particular example, both BLAST and INGA did not predict any correct class for this protein (the naïve root class is not counted here as it is trivial). When solely using global structure similarity (\(P_{\text{structure}}\)), we could only predict one single correct BP term. This makes sense because all the queries in our test set are difficult targets, which do not have close homologs in the template database. For instance, the closest template for this query P13342 is the methionine-tRNA ligase (the magenta structure in Figure 2.6(A)), which corresponds to the PDB ID 2CT8A. The sequence identity between P13342 and 2CT8A is only 6.84% and the TM-score between the two structures is only 0.24. Therefore, structure similarity or homology-based methods are not expected to predict the function of the query well. Structure information (\(P_{\text{structure}}\)) combined with PPI predicted three correct classes out of six. On the other hand, QAUST predicted four correct classes out of six. In addition, the prediction of QAUST is at least one level deeper in the GO hierarchy than the other methods. Meanwhile, the predicted MF and CC classes for this protein by QAUST are at least as accurate as other methods.
Figure 2.6: A study case for protein function prediction using QAUST. (A) The superimposition between the query (P13342, in cyan) and the closest template in the database (PDB ID 2CT8A, in magenta) based on the structural alignment generated by TM-align. (B) Predicted BP classes for protein P13342. The six BP classes (the root term, Biological Process, is a naive class, which is not counted) shown are the experimentally annotated classes. The colored contours represent the BP classes that are predicted by the corresponding methods.
2.3.4 Experimental validation of TRIM22 dimerization

To provide an experimental assessment of the performance of QAUST, we chose the human tripartite motif-containing 22 (TRIM22) protein as an example. TRIM22 is known as an interferon-inducible protein which shows antiviral activity, such as HIV, HBV and HCV [92, 93, 94]. Recent studies also showed that TRIM22 mediates autophagy in human macrophages[95]. However, the function of TRIM22 is still not comprehensively understood as the protein only exists in primates. With the help of our collaborators from SUSTech University, we applied QAUST to predict the function for TRIM22. Among the predicted GO classes with high consensus scores, some of the CC and BP classes agree well with the previously known functions of TRIM22, such as the CC class “nucleus” and the BP class “response to virus”. However, the only two predicted MF classes have quite high consensus scores, “protein binding” and “protein homodimerization activity”, suggesting that TRIM22 binds to itself to form a dimer. We thus set out to test if human TRIM22 can form homodimer using coimmunoprecipitation (Figure 2.7(A)). Our collaborators first expressed Flag or GFP-tagged human TRIM22 protein by co-transfecting the two plasmids into HEK293T cells. After 48-hour incubation, cells were harvested and lysed (20mM Tris-HCl pH 7.5, 150mMNaCl, 1mMEDTA, 1%NP – 40 with proteinase inhibitor). Both Flag and GFP-tagged TRIM22 were detected in cell lysate by western blot (Figure 2.6(B)). We then pulled down Flag-tagged TRIM22 from the cell lysate. For each sample, 25μl protein A/G beads were incubated with 1μg Flag antibody at 4 degree. Mouse IgG was used as a negative control. After 2 hours, beads were washed with lysis buffer and then incubated with 500μg cell lysate at 4 degree for another 2 hours. Western blot showed that when Flag-tagged TRIM22 was pulled down, GFP-tagged TRIM22 can be detected by GFP antibody (Figure 2.6(C)), which showed that GFP-tagged and Flag-tagged TRIM22 bind together in HEK293T. To further confirm this binding, we did co-IP in the opposite way. Flag-tagged TRIM22 was
also detected in immunoprecipitation of GFP-tagged TRIM22 (Figure 2.6D). These results reveal that TRIM22 can bind to itself, which is the most likely way to form the homodimer.

2.4 Discussion and limitations

In this first part of my thesis, I developed QAUST, a method to predict biological functions of protein molecules using three main features: global and local protein structure similarity, protein-protein interaction and functional sequence motifs. In my method, I constructed the 3D structure from the amino acid sequence using I-TASSER. Functional analogs are then identified by performing global and local structural similarity search through the functional libraries, with the scoring function involving the confidence score of structural predictions, sequence and structural similarity of the I-TASSER model with the functional templates, and the local active site matches. I have also tried to improve the performance of GO prediction by incorporating protein-protein interaction information, especially in order to improve the prediction of GO classes under BP and CC aspects. I further developed a novel predictor that extracts functional motifs that are related to a specific GO class and used it as our third predictor. On a set of 500 non-redundant proteins, QAUST is shown to have higher function prediction accuracy than other competing methods on most prediction tasks. This performance advantage is mainly a result of combining three different predictors which cover major aspects of proteins. Additionally, the three prediction components used complement each other in the sense that they contribute differently to the prediction of the three aspects of GO. While PPI information improves significantly the prediction of BP and CC classes, functional motifs detection is mainly useful in improving MF class prediction. However, QAUST has a number of limitations that give room for possible improvement in the future. One main limitation is that QAUST is much more expensive in terms of running time compared
Figure 2.7: Experimental validation of homodimerization function of TRIM22. (A) Illustration of the coimmunoprecipitation method to validate the homodimerization of TRIM22 performed by our collaborators at SUSTech. We expressed Flag- or GFP-tagged human TRIM22 protein by co-transfecting two plasmid into HEK293T cells. If TRIM22 forms a homodimer, when Flag-tagged TRIM22 or GFP-tagged TRIM22 is pulled down, both Flag-tagged and GFP-tagged TRIM22 should be detected by the corresponding antibodies (4 combinations in total). (B) Both Flag- and GFP-tagged TRIM22 expressed in HEK293T cells, detected by Western Blot. (C) For Flag-immunoprecipitation, both Flag-tagged and GFP-tagged TRIM22 are detected by the corresponding antibodies, whereas mouse IgG is used as a negative control. (D) For GFP-immunoprecipitation, both Flag-tagged and GFP-tagged TRIM22 are detected by the corresponding antibodies, whereas rabbit IgG is the negative control.
to the other methods. The second limitation is that the method proposed cannot be directly used to infer functions that are not included in EC or GO systems since it solely infers protein functions from existing protein annotations. Finally, given that the three components used work differently in predicting different aspects of GO, it may be helpful to weight their scores differently depending on the nature of the GO class evaluated instead of combining the scores in a simple consensus by using more sophisticated prediction models such as neural networks.

2.5 Assessment of protein GO function prediction and knowledge extraction from ontologies

A key question I had to continuously ask through this first part of my work is how to quantitatively measure the correctness of my predicted GO classes compared to the ground truth GO classes. This question is just a special case of the broader problem of extracting knowledge from ontologies. One way to extract knowledge from GO and other ontologies can be through the use of semantic similarity measures or other mathematical formulas like we did in this evaluation. However, these methods are very limited in the sense that they can only be applied on the subset of the ontology which can be represented in a graph structure. A substantial amount of knowledge encoded within ontologies is then inevitably missed by these measures. Another key limitation of such measures is that the information extracted will be encoded in one single numerical value that gives an estimation of the similarity between two ontology classes or two group of classes. This single numerical value can only tell us very little about the information contained within the ontology and cannot be used as a feature vector for machine learning models for example.

In the next part of my thesis which makes the key contribution of my doctoral work, I solve this problem by proposing the first methods that can encode all the formal and informal information within an ontology by learning concise and rich
vector representations of ontology classes as well as of the annotated entities.
Chapter 3

Knowledge representation from biomedical ontologies

3.1 Introduction

In this chapter, I propose Onto2Vec [39] and OPA2Vec [40], the first methods that can learn knowledge representations from all the formal (logical axioms) and informal (meta-data) content of biomedical ontologies. I show through diverse applications how the proposed methods can be used to accurately encode knowledge from ontologies and their annotations and how they can be used in combination with machine learning prediction methods for biomedical prediction and analysis tasks of interest.

Biomedical ontologies provide the means to formally structure the classes and relations within a domain, and are now employed by a wide range of biological databases, webservices, and file formats to provide semantic meta-data [2]. Notably, ontologies are used for the annotation of biological entities such as genomic variants, genes and gene products, or chemicals, to classify their biological activities and associations [96]. An annotation is an association of a biological entity (or a class of biological entities) and one or more classes from an ontology, usually together with meta-data about the source and evidence for the association, the author, etc. [97].

Due to the wide-spread use of ontologies, several methods have been developed to utilize the information in ontologies for data analysis [2]. As I discussed in the background chapter, a wide range of semantic similarity measures in particular has been developed [20] and applied to the similarity-based analysis of ontologies and entities annotated with them and to exploit information in ontologies
These measures have successfully been applied to the prediction of protein–protein interactions \[20\], gene–disease associations \[102\], or drug targets \[103\]. To be more specific, semantic similarity measures can be classified into different types depending on how annotations (or instances) of ontology classes are incorporated or weighted, and the type of information from an ontology that is used to determine the similarity \[20\ \[21\]. As discussed in the introduction, most similarity measures treat ontologies as graphs in which nodes represent classes and edges an axiom involving the connected classes \[20\ \[21\]. However, not all the axioms in an ontology can naturally be represented as graphs \[15\ \[25\ \[26\] which is one of the major limitations of such measures. The fact that the output of a similarity measure is limited to one single number is also one of the main reasons that machine learning cannot, or at least not optimally, utilize any of these traditional measures. In addition to similarity-based analysis, ontology-based annotations are frequently used in machine learning approaches. Ontology-based annotations can be encoded as binary vectors representing whether or not an entity is associated with a particular class, and the semantic content in ontologies (i.e., the subclass hierarchy) can be used to generate “semantically closed” feature vectors \[104\]. Alternatively, the output of semantic similarity measures is widely used as features for machine learning applications, for example in drug repurposing systems \[105\] or identification of causative genomic variants \[106\ \[107\]. These approaches have in common that the features generated through them contain no explicit information about the structure of the ontology and therefore of the dependencies between the different features; these dependencies are therefore no longer available as features for a machine learning algorithm. In the case of semantic similarity measures, the information in the ontology is used to define the similarity but the information used to define the similarity is subsequently reduced to a single point (the similarity value); in the case of binary feature vectors, the ontology structure is used to generate the values of the feature vector but is subsequently
no longer present or available to a machine learning algorithm. Feature vectors that explicitly encode for both the ontology structure and an entity’s annotations would contain more information than either information alone and may perform significantly better in machine learning applications than alternative approaches.

Finally, semantic similarity measures are generally hand-crafted, i.e., they are designed by an expert based on a set of assumptions about how an ontology is used and what should constitute a similarity. However, depending on the application of semantic similarity, different features may be more or less relevant to define the notion of similarity. It has previously been observed that different similarity measures perform well on some datasets and tasks, and worse on others [108, 109, 20, 110], without any measure showing clear superiority across multiple tasks. One possible way to define a common similarity measure that performs equally well on multiple tasks may be to establish a way to train a semantic similarity measure in a data-driven way. While this is not always possible due to the absence of training data, when a set of desired outcomes (i.e., labelled data points) are available, such an approach may result in better and more intuitive similarity measures than hand-crafted ones.

In addition to these traditional measures, a set of methods have been developed that can characterize nodes and edges in knowledge graphs through “embeddings”. A knowledge graph is a directed graph which consists of nodes that represent entities within a domain of knowledge, labeled edges which represent relations between these entities, and an inference mechanisms that enables the generation of new relations between entities in the graph. A knowledge graph embedding is a function that maps entities (nodes and edges) in a knowledge graph to vectors within an $n$-dimensional vector space subject to constraints that aim to preserve certain structural features of the graph within the vector space; several methods to generate knowledge graph embeddings have been developed that primarily differ in the constraints they employ on the mapping function [27, 28, 30]. These methods are used to produce feature
vectors for entities represented in a knowledge graph and encode for (parts of) the knowledge about the entity that is represented in a knowledge graph. Knowledge graph embeddings have already been applied successfully in the biological domain to predict relations between biological entities \[35, 111\]. However, ontologies, in particular those in the biomedical domain, cannot easily be represented as graphs \[112\]; rather, they constitute logical theories that are best represented as sets of axioms \[113\].

In the first part of this chapter on knowledge representation from ontologies, I propose Onto2Vec, a novel method to jointly produce dense vector representations of biological entities, their ontology-based annotations, and the ontology structure used for annotations \[39\]. In this part of my thesis, I apply the method to the Gene Ontology (GO) \[3\] and generate dense vector representations of proteins and their GO annotations. I demonstrate that Onto2Vec generates vectors that can outperform traditional semantic similarity measures in the task of similarity-based prediction of protein-protein interactions; I also show how to use Onto2Vec to train a semantic similarity measure in a data-driven way, and use this to predict protein-protein interactions and distinguish between the types of interactions. I further apply Onto2Vec-generated vectors to clustering and show that the generated clusters reproduce Enzyme Commission numbers of proteins. The Onto2Vec method is generic and can be applied to any set of entities and their ontology-based annotations, and the implementation is freely available at https://github.com/bio-ontology-research-group/onto2vec.

In the second part of this chapter, I extend Onto2Vec to OPA2Vec (Ontologies Plus Annotations to Vectors) to jointly produce vector representations of entities in biomedical ontologies based on both the semantic content of ontologies (i.e., the logical axioms) and the meta-data contained in ontologies as Web Ontology Language (OWL) \[114, 12\] annotation axioms. I combine multiple types of information contained in biomedical ontologies, including asserted and inferred logical axioms,
datatype properties, and annotation axioms to generate a corpus that consists of both formal statements, natural language statements, and other annotation axiom values that relate entities to literals. I then apply a skip-gram neural language model to generate vector representations for any entity named in the ontology. I further extend the method by incorporating information from biomedical literature. Using transfer learning, I apply a pre-trained Word2Vec model in OPA2Vec to significantly improve the performance in encoding natural language phrases and statements.

I evaluate OPA2Vec using two different ontologies and applications: first, I use the Gene Ontology (GO) [3] to produce vector representations of yeast and human proteins and determine their functional similarity and predict interactions between them; second, I evaluate the method on the PhenomeNET ontology [41, 115] to infer vector representations of genes and diseases and use them to predict gene–disease associations. I demonstrate that OPA2Vec can produce task-specific and trainable representations of biological entities that significantly outperform both Onto2Vec and traditional semantic similarity measures in predicting protein–protein interactions and gene–disease associations. OPA2Vec is a generic method which can be applied to any ontology formalized in OWL, and OPA2Vec is freely available from https://github.com/bio-ontology-research-group/opa2vec/.

3.2 Materials and Methods

3.2.1 Ontology and annotation resources

The Gene Ontology (GO)

I used the Gene Ontology (GO) [3] in OWL format from http://www.geneontology.org/ontology/ used in the previous chapter and that has been obtained on September 13, 2017. I obtained the GO-protein annotations from the UniProt-GOA website (http://www.ebi.ac.uk/GOA) on September 26, 2017. I removed all annotations
with evidence code IEA as well as ND. For validation, I used the STRING database [116] to obtain protein–protein interaction (PPI) data for human (*Homo sapiens*) and yeast (*Saccharomyces cerevisiae*), downloaded on September 16, 2017. The yeast PPI network contains 2,007,135 interactions with 6,392 unique proteins, while the human PPI network contains 11,353,057 interactions for 19,577 unique proteins.

**PhenomeNET ontology**

I downloaded the PhenomeNET ontology [11,115] in owl format from the AberOWL repository [http://aber-owl.net] [117] on February 21, 2018. I downloaded the mouse phenotype annotations from the Mouse Genome Informatics (MGI) database [http://www.informatics.jax.org/] [118] on February 21, 2018. I obtained a total of 302,013 unique mouse phenotype annotations. I obtained the disease to human phenotype annotations on February 21, 2018 from the Human Phenotype Ontology (HPO) database [http://human-phenotype-ontology.github.io/] [119]. I downloaded only the OMIM disease to human phenotype annotations which resulted in a total of 78,208 unique disease-phenotype associations. For gene–disease association prediction validation, I used the MGI.DO.rpt file from the MGI database. This file contains 9,506 mouse gene-OMIM disease associations and 13,854 human gene-OMIM disease associations. To map mouse genes to human genes I used the HMD_HumanPhenotype.rpt file from the MGI database; the mapping between mouse and human genes is necessary because gene–disease associations are reported for human genes (in one of the evaluation sets) while the phenotypes and phenotype-based predictions are made for mouse genes.
3.2.2 Data sets

Protein interaction network

To predict protein interactions, I used 1,015 proteins from the yeast data set for training and 677 randomly selected proteins for testing while I used 2,263 proteins from the human data set for training and 1,509 for testing. I considered as positives the pairs in the STRING database [116] and I randomly sub-sampled negatives among all the pairs not occurring in STRING; I ensure that the cardinality of the positives and negatives are equal for the testing and the training datasets.

Gene–disease associations

To predict gene–disease associations observed in mouse models, I used 6,710 gene–disease associations for training (2,030 diseases and all their associations) and 2,876 for testing (870 diseases and all their associations); for gene–disease associations observed in humans, I used 9,698 associations for training (2,978 diseases) and 4,196 for testing (1,276 diseases). I used the gene–disease associations from the MGI.DO.rpt available at MGI; I consider all other associations as negatives.

I evaluate and compare all methods on the same testing data that was obtained through the random selection: 667 yeast proteins (and all their interactions), 1,509 human proteins (and all their interactions), 870 diseases from gene–disease associations in mice, and 1,276 diseases gene–disease associations in human.

3.2.3 Automated reasoning

I used the OWL API version 4.2.6 [120] to process the ontologies in OWL format [121]. The version of GO I use contains 577,454 logical axioms and 43,828 classes. I used the HermiT reasoner (version 1.3.8.413) [122] to infer new logical axioms from the asserted ones from GO. I used HermiT as it supports all OWL 2 DL axioms and has
been optimized for large ontologies \cite{122}. These optimizations make HermiT relatively fast which is particularly helpful when dealing with ontologies of the size of GO. I infer three types of axioms: subsumption, equivalence and disjointness, resulting in 80,133 new logical axioms that are implied by GO’s axioms and materialized through HermiT. For reasoning on PhenomeNET, I used Elk reasoner \cite{123} which is more suitable for bigger ontologies.

3.2.4 Representation learning using Word2Vec

I treated an ontology as a set of axioms, each of which constitutes a sentence. To process the axioms syntactically, I used the Word2Vec \cite{124,125} methods. Word2Vec is a set of neural-network based tools which generate vector representations of words from large corpora. The vector representations are obtained in such a way that words with similar contexts tend to be close to each other in the vector space. Word2Vec can use two distinct models: the continuous bag of word (CBOW), which uses a context to predict a target word, and the skip-gram model which tries to maximize the classification of a word based on another word from the same sentence. The main advantage of the CBOW model is that it smooths over a lot of the distributional information by treating an entire context as one observation, while the skip-gram model treats each context-target as a new observation, which works better for larger datasets. The skip-gram model has the added advantage of producing higher quality representation of rare words in the corpus \cite{124,125}. Here, I chose the skip-gram architecture since it meets the need to produce high quality representations of all biological entities occurring in the large corpus, including infrequent ones. Formally, given a sequence of training words $\omega_1, \omega_2, ..., \omega_T$, the skip-gram model aims to maximize the following average log likelihood:

$$
\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(\omega_{t+j}|\omega_t),
$$

(3.1)
where $c$ is the size of the training context, $T$ is the size of the set of the training words and $\omega_i$ is the $i$-th training word in the sequence. I identified an optimal set of parameters of the skip-gram model through limited gridsearch on the following parameters: the size of the output vectors on the interval $[50-250]$ using a step size of 50, the number of iterations on the interval $[3-5]$ and negative sampling on the interval $[2-5]$ using a step size of 1. Table 3.1 shows the parameter values I used for the skip-gram in this work.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$sg$</td>
<td>Choice of training algorithm</td>
<td>$sg = 1$ skip-gram $sg = 0$ CBOW 1</td>
</tr>
<tr>
<td>size</td>
<td>Dimension of the obtained vectors</td>
<td>200</td>
</tr>
<tr>
<td>$min_count$</td>
<td>Words with frequency lower than this value will be ignored</td>
<td>1</td>
</tr>
<tr>
<td>window</td>
<td>Maximum distance between the current and the predicted word</td>
<td>10</td>
</tr>
<tr>
<td>iter</td>
<td>Number of iterations</td>
<td>5</td>
</tr>
<tr>
<td>negative</td>
<td>Whether negative sampling will be used and how many “noise words” would be drawn</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3.1: Parameters used for training the Word2Vec model.

### 3.2.5 Text corpora

I retrieved the entire collection of article abstracts in the MEDLINE format from the PubMed database [https://www.ncbi.nlm.nih.gov/pubmed/] on February 6, 2018. The total number of abstracts collected is 28,189,045. For each abstract, I removed the meta-data (publication date, journal, authors, PMID, etc.), and only kept the title of the article and the text of the abstract for training a Word2Vec model.
PubMed Central (PMC) is a repository provided by the NCBI containing full texts of peer-reviewed journal articles in the life sciences. We have downloaded all the open-access PMC articles on June 10, 2018 which resulted in a total of 4,985,333 full-text articles. For OPA2Vec, I used these articles to pre-train a Word2Vec model that can be compared to the model trained on Medline.

3.2.6 Similarity

I used cosine similarity to determine similarity between feature vectors generated by my learning methods. The cosine similarity, $cos_{sim}$, between two vectors $A$ and $B$ is calculated as

$$cos_{sim}(A, B) = \frac{A \cdot B}{||A|| ||B||},$$

where $A \cdot B$ is the dot product of $A$ and $B$.

I used Resnik’s semantic similarity measure [22] as the baseline for comparison. Resnik’s semantic similarity measure is widely used in biology [20]. It is based on the notion of information content (IC) which quantifies the specificity of a given class in the ontology. The information content of a class $c$ is defined as the negative log likelihood, $-\log p(c)$, where $p(c)$ is the probability of encountering an instance or annotation of class $c$. Given this definition of information content, Resnik similarity is formally defined as:

$$sim_{Resnik}(c_1, c_2) = -\log p(c_{MICA}),$$

where $c_{MICA}$ is the most informative common ancestor of $c_1$ and $c_2$ in the ontology hierarchy, defined as the common ancestor of $c_1$ and $c_2$ with the highest information content value. In addition to Resnik’s similarity I also compare to three other semantic similarity measures: Lin’s measure [23], Jiang&Conrath measure [24] and $sim_{GIC}$.
Lin’s similarity measure is defined as:

\[
\text{sim}_{\text{Lin}}(c_1, c_2) = \frac{2 \cdot \log p(c_{\text{MICA}})}{\log p(c_1) + \log p(c_2)},
\]

(3.4)

Jiang&Conrath similarity (\(\text{sim}_{\text{J&C}}\)) uses the same components used in Lin’s similarity but with a different formulation:

\[
\text{sim}_{\text{J&C}}(c_1, c_2) = 2 \cdot \log p(c_{\text{MICA}}) - \log p(c_1) - \log p(c_2),
\]

(3.5)

\text{sim}_{\text{GIC}} measure is different than the three previously defined measures in the sense that it calculates the similarity between entities instead of concepts. Given entities \(e_1\) and \(e_2\), their pairwise similarity according to \(\text{sim}_{\text{GIC}}\) is the following:

\[
\text{sim}_{\text{GIC}}(e_1, e_2) = \frac{\sum_{c \in e_1 \cap e_2} - \log p(c)}{\sum_{c \in e_1 \cup e_2} - \log p(c)},
\]

(3.6)

where \(e_1 \cap e_2\) is the set of ontology concepts that both A and B are annotated with, while \(e_1 \cup e_2\) is the union of all concepts that A and B are annotated with (not just the shared concepts). Resnik’s, Lin’s and Jiang&Conrath similarity measures only measure the similarity between two ontology classes. A protein (or other biomedical entities) can be involved in different biological processes and can carry several molecular functions, it can be annotated with more than one GO term. Therefore, to calculate semantic similarity between a pair of proteins, or a pair of any biological entities, it is necessary to properly aggregate the similarity between the concepts that they are respectively annotated with. One possible way to achieve that would be to calculate the Best Match Average (BMA) which estimates the average similarity between the best matching terms of two concepts [127]. For two biological entities \(e_1\)
and \( e_2 \), the BMA is defined as:

\[
BMA(e_1, e_2) = \frac{1}{2} \left( \frac{1}{n} \sum_{c_1 \in S_1} \max_{c_2 \in S_2} \text{sim}(c_1, c_2) + \frac{1}{m} \sum_{c_2 \in S_2} \max_{c_1 \in S_1} \text{sim}(c_1, c_2) \right),
\]

(3.7)

where \( S_1 \) is the set of ontology concepts that \( e_1 \) is annotated with, \( S_2 \) is the set of concepts that \( e_2 \) is annotated with, and \( \text{sim}(c_1, c_2) \) is the similarity value between concept \( c_1 \) and concept \( c_2 \), which could have been calculated using Resnik similarity or any other semantic similarity measure (e.g., cosine similarity).

### 3.2.7 Supervised learning

I used supervised learning to train a similarity measure between two entities that is predictive of their pairwise interactions or associations.

#### Evaluation data sets

I apply the proposed methods to two datasets, one for protein interactions in yeast and another in human and for gene–disease associations on human and mouse.

For the protein interaction prediction, I used 1,015 proteins from the yeast data set for training and 677 randomly selected proteins for testing while I used 2,263 proteins from the human data set for training and 1,509 for testing. I considered as positives the pairs in the STRING database and I randomly sub-sampled negatives among all the pairs not occurring in STRING; I ensure that the cardinality of the positives and negatives are equal for the testing and the training datasets.

When predicting gene–disease associations observed in mouse models, I used 6,710 gene–disease associations for training (2,030 diseases and all their associations) and 2,876 for testing (870 diseases and all their associations); for gene–disease associations observed in humans, I used 9,698 associations for training (2,978 diseases) and 4,196 for testing (1,276 diseases). I used the gene–disease associations from the MGI_DO.rpt
available at MGI; I consider all other associations as negatives.

I evaluate and compare all methods on the same testing data that we obtained through the random selection: 667 yeast proteins (and all their interactions), 1,509 human proteins (and all their interactions), 870 diseases from gene–disease associations in mice, and 1,276 diseases gene–disease associations in human.

**Classification methods**

I used logistic regression, support vector machines (SVMs), and artificial neural networks (ANNs) to train a classifier for protein-protein interactions. I trained each of these methods by providing a pair of proteins (represented through their feature vectors) as input and predicting whether the pair interacts or not. The output of each method varies between 0 and 1, and I used the prediction output as a similarity measure between the two inputs. Logistic regression does not require any selection of parameters. I used the SVM with a linear kernel and sequential minimal optimization.

I chose a deep multi-layer perceptron neural network to be a feed-forward network with three hidden layers. The architecture of the network is shown in Figure 3.1. I optimized parameters using a limited manual search based on best practice guidelines [128]. I optimized the ANN using binary cross entropy as the loss function.

**Figure 3.1: Architecture of the neural network used for the classification**
3.2.8 Clustering and visualization

For visualizing the ontology vectors I generated, I used the t-SNE method to reduce the dimensionality of the vectors to 2 dimensions, and plotted the vectors in the 2D space. t-SNE is similar to principal component analysis but uses probability distributions to capture the non-linear structure of the data points, which linear dimensionality reduction methods, such as PCA, cannot achieve. I used a perplexity value of 30 when applying t-SNE. The k-means algorithm is used to cluster the protein vectors, and I quantitatively measured the quality of these clusters with respect to EC families by using cluster purity. Cluster purity is defined as:

$$purity(T, C) = \frac{1}{N} \sum_{i=0}^{k} \max_j (c_k \cup t_j),$$  \hspace{1cm} (3.8)

where $N$ is the total number of data points, $C = c_1, c_2, ..., c_k$ is the set of clusters, and $T = t_1, t_2, ..., t_J$ is the set of classes which is in this case the set of EC families. Since there are six first-level EC categories, the number of classes in this case is six and the number of clusters used in k-means is also set to six.

3.2.9 Text-mining based prediction method

I compare OPA2Vec to the text-mining based prediction method BeFree. BeFree extracts sets of biological associations from scientific articles. I downloaded the BeFree gene–disease prediction from the DisGeNet database on September 30, 2018. The BeFree gene–disease associations are represented using UMLS concept identifiers. I use the Disease Ontology (DO) to map the UMLS identifiers to OMIM identifiers. Since not all diseases have both an OMIM and a UMLS identifier, I use a limited evaluation set consisting of 1,194 diseases shared between BeFree and the evaluation set.
3.2.10 Evaluation using ROC and AUC

The Receiver Operating Characteristic (ROC) curve is a widely used evaluation method to assess the performance of prediction and classification models. It plots the true-positive rate (TPR or sensitivity) defined as \( TPR = \frac{TP}{TP + FN} \) against the false-positive rate (FPR or 1−specificity) defined as \( FPR = \frac{FP}{FP + TN} \), where \( TP \) is the number of true positives, \( FP \) is the number of false positives and \( TN \) is the number of true negatives \[136\]. I used ROC curves to evaluate the prediction performance of my methods as well as the performance of the baseline methods. In the evaluation, the \( TP \) value is the number of protein pairs occurring in STRING regardless of their STRING confidence score and which have been predicted as interacting. The \( FP \) value is the number of protein pairs which have been predicted as interacting but do not appear in STRING. And the \( TN \) is the number of protein pairs predicted as non-interacting and which do not occur in the STRING database. In most cases, ROC curves of different methods would most probably overlap which makes the visual test of the ROC curves insufficient to make a formal comparison between different methods \[137\]. This is the reason why there is need for a quantitative measure that summarizes the meaning of a ROC curve and allows to more formally compare different methods. The most popular of these measures is the area under the ROC curve (AUC) which is the integration of the ROC curve over the entire FPR axis \[137\]. To evaluate the predictive performance of both Onto2Vec and OPA2Vec for protein interactions and gene–disease associations, the AUC has also been used along with the ROC curve.

3.3 Onto2Vec: Knowledge representation from the formal content of ontologies

With my supervisors, I developed Onto2Vec, a method to learn dense, vector-based representations of classes in ontologies, and the biological entities annotated with
classes from ontologies. To generate the vector representations, we combined symbolic inference (i.e., automated reasoning) and statistical representation learning. We first generated vector-based representations of the classes in an ontology, and then extended the result to generate representations of biological entities annotated with these classes. The vector-based representations generated by Onto2Vec provide the foundation for machine learning and data analytics applications, including semantic similarity applications.

The main contribution with Onto2Vec is a method to learn a representation of individual classes (and other entities) in an ontology, taking into account all the axioms in an ontology that may contribute to the semantics of a class, either directly or indirectly. Onto2Vec uses an ontology \( O \) in the OWL format, and applies the HermiT OWL reasoner [122] to infer new logical axioms, i.e., equivalent class axioms, subclass axioms, and disjointness axioms (for technical details on automated reasoning see Section 3.2.3). As an example, if axiom \( A \ SubClassOf B \) and axiom \( B \ SubClassOf C \) both occur in the original set of ontology axioms, the reasoner can infer axiom \( A \ SubClassOf C \) which correctly describes the ontology, but does not exist in the initial set of the ontology axioms. We call the union of the set of axioms in \( O \) and the set of axioms derived using the reasoner the deductive closure of \( O \), designated \( O^+ \). In contrast to treating ontologies as taxonomies or graph-based structures [26], we assume that every axiom in \( O \) (and consequently in \( O^+ \)) constitutes a sentence, and the set of axiom in \( O \) (and \( O^+ \)) a corpus of sentences. The vocabulary of this corpus consists of the classes and relations that occur in \( O \) as well as the keywords used to formulate the OWL axioms [12, 114]. Onto2Vec then uses a skip-gram model to learn a representation of each word that occurs in the corpus. The representation of a word in the vocabulary (and therefore of a class or property in \( O \)) is a vector that is predictive of words occurring within a context window [124, 125] (see Section 3.2.4 for details).
Onto2Vec can also be used to learn vector-based representations of biological entities that use ontologies for annotation and combine information about the entities’ annotations as well as the semantics of the classes used in the annotation in a single representation. Trivially, since Onto2Vec can generate representations of single classes in an ontology, an entity annotated with $n$ classes, $C_1, ..., C_n$, can be represented as a (linear) combination of the vector representations of these classes. For example, if an entity $e$ is annotated with $C_1$ and $C_2$, and $\nu(C_1)$ and $\nu(C_2)$ are the representations of $C_1$ and $C_2$ generated through Onto2Vec, I can use $\nu(C_1) + \nu(C_2)$ as a representation of $e$. Alternatively, we can use Onto2Vec directly to generate a representation of $e$ by extending the axioms in $O$ with additional axioms that explicitly capture the semantics of the annotation. If $O'$ is the ontology generated from annotations of $e$ by adding new axioms capturing the semantics of the annotation relation to $O$, then $e$ is a new class or instance in $O'$ for which Onto2Vec will generate a representation (since $e$ will become a word in the corpus of axioms generated from $O'^\pi$).

As comprehensive use case, we applied the method to the GO, and to a joint knowledge base consisting of GO and proteins with manual GO annotations obtained from the UniProt database. To generate the latter knowledge base, we added proteins as new entities and connected them using a has-function relation to their functions. We then applied Onto2Vec to generate vector representations for each class in GO (using a corpus based only on the axioms in GO), and further generate joint representations of proteins and GO classes (using a corpus based on the axioms in GO and proteins, and their annotations). We further generated protein representations by combining (i.e., adding) the GO class vectors of the proteins’ GO annotations (i.e., if a protein $p$ is annotated to $C_1, ..., C_n$ and $\nu(C_1), ..., \nu(C_2)$ are the Onto2Vec-vectors generated for $C_1, ..., C_n$, we define the representation $\nu(p)$ of $p$ as $\nu(p) = \nu(C_1) + ... + \nu(C_n)$). In total, we generated 556,388 vectors representing proteins (each protein is represented three times, either as a set of GO class vectors, the sum of GO class vectors, or a
Figure 3.2: Onto2Vec Workflow. The blue-shaded part illustrates the steps to obtain vector representation for classes from the ontology. The purple-shaded part shows the steps to obtain vector representations of ontology classes and the entities annotated to these classes.

vector jointly generated from representing has-function relations in the knowledge base), and 43,828 vectors representing GO classes. Figure 3.2 illustrates the main Onto2Vec workflow to construct ontology-based vector representations of classes and entities.

### 3.3.1 Similarity-based prediction of biological relations

<table>
<thead>
<tr>
<th></th>
<th>Yeast</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik</td>
<td>0.1107</td>
<td>0.1151</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.1067</td>
<td>0.1099</td>
</tr>
<tr>
<td>Binary.GO</td>
<td>0.1021</td>
<td>0.1031</td>
</tr>
<tr>
<td>Onto2Vec_LR</td>
<td>0.1424</td>
<td>0.1453</td>
</tr>
<tr>
<td>Onto2Vec_SVM</td>
<td>0.2245</td>
<td>0.2621</td>
</tr>
<tr>
<td>Onto2Vec_NN</td>
<td>0.2516</td>
<td>0.2951</td>
</tr>
<tr>
<td>Binary.GO_LR</td>
<td>0.1121</td>
<td>0.1208</td>
</tr>
<tr>
<td>Binary.GO_SVM</td>
<td>0.1363</td>
<td>0.1592</td>
</tr>
<tr>
<td>Binary.GO_NN</td>
<td>0.1243</td>
<td>0.1616</td>
</tr>
</tbody>
</table>

Table 3.2: Spearman correlation coefficients between STRING confidence scores and PPI prediction scores of different prediction methods. The highest absolute correlation across all methods is highlighted in bold.
<table>
<thead>
<tr>
<th>Method</th>
<th>Yeast</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik</td>
<td>0.7942</td>
<td>0.7891</td>
</tr>
<tr>
<td>Lin</td>
<td>0.7354</td>
<td>0.7222</td>
</tr>
<tr>
<td>Jiang&amp;Conrath</td>
<td>0.7108</td>
<td>0.7027</td>
</tr>
<tr>
<td>sim_GIC</td>
<td>0.7634</td>
<td>0.7594</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.7701</td>
<td>0.7614</td>
</tr>
<tr>
<td>Onto2Vec_NoReasoner</td>
<td>0.7439</td>
<td>0.7385</td>
</tr>
<tr>
<td>Binary.GO</td>
<td>0.6912</td>
<td>0.6712</td>
</tr>
<tr>
<td>Onto_BMA</td>
<td>0.6741</td>
<td>0.6470</td>
</tr>
<tr>
<td>Onto_AddVec</td>
<td>0.7139</td>
<td>0.7093</td>
</tr>
<tr>
<td>Onto2Vec_LR</td>
<td>0.7959</td>
<td>0.7785</td>
</tr>
<tr>
<td>Onto2Vec_SVM</td>
<td>0.8586</td>
<td>0.8621</td>
</tr>
<tr>
<td>Onto2Vec_NN</td>
<td>0.8869</td>
<td>0.8931</td>
</tr>
<tr>
<td>Binary.GO_LR</td>
<td>0.7009</td>
<td>0.7785</td>
</tr>
<tr>
<td>Binary.GO_SVM</td>
<td>0.8253</td>
<td>0.8068</td>
</tr>
<tr>
<td>Binary.GO_NN</td>
<td>0.7662</td>
<td>0.7064</td>
</tr>
</tbody>
</table>

Table 3.3: AUC values of ROC curves for PPI prediction. The best AUC value among all methods is shown in bold. Resnik, Lin, Jiang&Conrath and sim_GIC are semantic similarity measures; Onto2Vec is my method in which protein and ontology class representations are learned jointly from a single knowledgebase which is deductively closed; Onto2Vec_NoReasoner is identical to Onto2Vec but does not use the deductive closure of the knowledge base; Binary.GO represents a protein’s GO annotations as a binary vector (closed against the GO structure); Onto_BMA only generates vector representations for GO classes and compares proteins by comparing their GO annotations individually using cosine similarity and averaging individual values using the Best Match Average approach; Onto_AddVec sums GO class vectors to represent a protein. The methods with suffix LR, SVM, and NN use logistic regression, a support vector machine, and an artificial neural network, respectively, either on the Onto2Vec or the Binary.GO protein representations.

We applied the vectors generated for proteins and GO classes to the prediction of protein-protein interactions by functional, semantic similarity. Onto2Vec also takes several parameters, including whether to deductively close the ontology, what size the embedding should be, and all parameters used in training the Word2Vec model; we use a gold-standard set of protein-protein interactions to determine the optimal set of these parameters. As a first experiment, we evaluated the accuracy of Onto2Vec in predicting protein-protein interactions. For this purpose, we generated several representations of proteins: first, we used Onto2Vec to learn representations of proteins.
jointly with representations of GO classes by adding proteins and their annotations to the GO using the has-function relations; second, we represented proteins as the sum of the vectors representing the classes to which they are annotated; and third, we represented proteins as the set of classes to which they are annotated.

We used cosine similarity to determine the similarity between vectors. To compare sets of vectors (representing GO classes) to each other, we used the Best Match Average (BMA) approach [20], where pairs of vectors are compared using cosine similarity. We term the approach in which we compared vectors generated from adding proteins to the knowledge base Onto2Vec; Onto_AddVec when using cosine similarity between protein vectors generated by adding the vectors of the GO classes to which they annotated; and Onto_BMA when using the BMA approach to compare sets of GO classes. To compare the different approaches for using Onto2Vec to the established baseline methods, we further applied the Resnik’s semantic similarity measure [22] with the BMA approach, and we generated sparse binary vector representations from proteins’ GO annotations [104] and compared them using cosine similarity (termed Binary_GO). In addition to Resnik, we also compare to three additional semantic similarity measures: Lin’s measure [23] and Jiang&Conrath’s measure [24] combined with the BMA approach as well as the sim_GIC’s similarity measure [126]. Furthermore, to evaluate the contribution of using an automated reasoner to infer axioms, we also included the results of using the Onto2Vec approach without applying a reasoner. The similarity measures we employed are formally described in Section 3.2.6.

We evaluated the performance of the method using protein-protein interaction datasets in two species, human (H. sapiens) and baker’s yeast (S. cerevisiae). Figure 3.3 shows the ROC curves obtained for each approach on the human and the yeast datasets; the area under the ROC curve (ROCAUC) values are shown in Table 3.3 (for details on how the evaluation was performed, see Section 3.2.10). With the exception of Resnik’s measure, we found that the other semantic similarity measures
Figure 3.3: ROC curves for PPI prediction for the unsupervised learning methods perform worse than the Onto2Vec model. Therefore, for the sake of conciseness we do only report the ROC curve of Resnik and not the other three measures. However, the ROCAUC values of all measures are reported in Table 3.3. We found that Resnik’s semantic similarity measure performs better than all other methods we evaluated, and that the Onto2Vec representation based on generating representations jointly from proteins and GO classes performs second best. These results demonstrate that Resnik’s semantic similarity measure, which determines similarity based on the information content of ontology classes as well as the ontology structure, is better suited for this application than the Onto2Vec representations using cosine similarity.

However, a key feature of Onto2Vec representations is their ability to encode for annotations and the ontology structure; while cosine similarity (and the derived measures) can determine whether two proteins are similar, certain classes and axioms may contribute more to predicting protein-protein interactions than others. To test whether we can use the information in Onto2Vec representations in such a way, we used supervised machine learning to train a similarity measure that is predictive of protein-protein interactions. To this end, we used three different machine learning...
methods, logistic regression, support vector machines (SVMs), and neural networks (see Section 3.2.7 for details). To obtain a baseline comparison, we also trained each model using the Binary_GO protein representations.

Each model uses a pair of protein vectors as inputs and is trained to predict whether the proteins provided as input interact or not. Each supervised model also outputs intermediate confidence values and can therefore be considered to output a form of similarity. The ROC curves of all trained models using the Onto2Vec and binary representations of proteins are shown in Figure 3.4, and their ROCAUC values are reported in Table 3.3. We observed that the supervised models (i.e., the “trained” semantic similarity measures) using Onto2Vec protein representations outperform the use of pre-defined similarity measures in all experiments; while logistic regression performs comparable to Resnik semantic similarity, both SVMs and artificial neural networks can learn similarity measures that predict protein-protein interactions significantly better than any pre-defined similarity measure. Onto2Vec representations further outperform the sparse binary representations of protein functions, indicating that the combination of annotations and ontology axioms indeed results in improved predictive performance.

We further tested whether the supervised models (i.e., the trained semantic similarity measures) can be used as similarity measures so that higher similarity values represent more confidence in the existence of an interaction. We used the confidence scores associated with protein-protein interactions in the STRING database and determined the correlation between the prediction score of the trained models produced and the confidence score in STRING. Table 3.4 summarizes the Spearman correlation coefficients for each of the methods we evaluated. We found that the trained similarity measures correlate more strongly with the confidence measures provided by STRING than other methods, thereby providing further evidence that Onto2Vec representations encode useful information that is predictive of protein-protein inter-
Figure 3.4: ROC curves for PPI prediction for the supervised learning methods, in addition to Resnik’s semantic similarity measure for comparison.

<table>
<thead>
<tr>
<th>Method</th>
<th>Yeast</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik</td>
<td>0.1107</td>
<td>0.1151</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.1067</td>
<td>0.1099</td>
</tr>
<tr>
<td>Binary_Go</td>
<td>0.1021</td>
<td>0.1031</td>
</tr>
<tr>
<td>Onto2Vec_LR</td>
<td>0.1424</td>
<td>0.1453</td>
</tr>
<tr>
<td>Onto2Vec_SVM</td>
<td>0.2245</td>
<td>0.2621</td>
</tr>
<tr>
<td>Onto2Vec_NN</td>
<td><strong>0.2516</strong></td>
<td><strong>0.2951</strong></td>
</tr>
<tr>
<td>Binary_Go_LR</td>
<td>0.1121</td>
<td>0.1208</td>
</tr>
<tr>
<td>Binary_Go_SVM</td>
<td>0.1363</td>
<td>0.1592</td>
</tr>
<tr>
<td>Binary_Go_NN</td>
<td>0.1243</td>
<td>0.1616</td>
</tr>
</tbody>
</table>

Table 3.4: Spearman correlation coefficients between STRING confidence scores and PPI prediction scores of different prediction methods. The highest absolute correlation across all methods is highlighted in bold.

Finally, we trained the models to separate protein-protein interactions into different interaction types, as classified by the STRING database: reaction, activation, binding, and catalysis. For comparison, we also reported results when using sparse binary representations of proteins in the supervised models, and we reported Resnik semantic similarity and Onto2Vec similarity results (using cosine similarity). Table 3.5 and Table 3.6 summarize the results for human and yeast respectively. While
Resnik semantic similarity and Onto2Vec similarity cannot distinguish between different types of interaction, we find that the supervised models, in particular the multiclass SVM and artificial neural network, are capable when using Onto2Vec vector representations to distinguish between different types of interaction. In addition, the Onto2Vec representations perform better than sparse binary vectors, indicating further that encoding parts of the ontology structure can improve predictive performance.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reaction</td>
<td>Activation</td>
</tr>
<tr>
<td>Resnik</td>
<td>0.5341</td>
<td>0.5331</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.5153</td>
<td>0.5104</td>
</tr>
<tr>
<td>Onto2Vec_LR</td>
<td>0.7091</td>
<td>0.6951</td>
</tr>
<tr>
<td>Onto2Vec_multiSVM</td>
<td>0.7351</td>
<td>0.7583</td>
</tr>
<tr>
<td>Onto2Vec_NN</td>
<td>0.7265</td>
<td>0.7568</td>
</tr>
<tr>
<td>Binary.GO_LR</td>
<td>0.6151</td>
<td>0.6533</td>
</tr>
<tr>
<td>Binary.GO_multiSVM</td>
<td>0.7246</td>
<td>0.7132</td>
</tr>
<tr>
<td>Binary.GO_NN</td>
<td>0.6895</td>
<td>0.6803</td>
</tr>
</tbody>
</table>

Table 3.5: AUC values of the ROC curves for PPI interaction type prediction for the **human** dataset. The best AUC value for each action is shown in bold.

Table 3.6: AUC values of the ROC curves for PPI interaction type prediction for the **yeast** dataset. The best AUC value for each action is shown in bold.
3.3.2 Clustering and visualization

Onto2Vec representations can not only be used to compute semantic similarity or form part of supervised models, but can also provide the foundation for visualization and unsupervised clustering. The ability to identify sets of biological entities which are more similar to each other within a dataset can be used for clustering and identifying groups of related biological entities. We visualized the GO-based vector representations of proteins generated by Onto2Vec. Since the Onto2Vec representations are of a high dimensionality, we applied the t-SNE dimensionality reduction [129] to the vectors and represented 10,000 randomly chosen enzyme proteins in Figure 3.5 (see Section 3.2.8 for details).

The visual representation of the enzymes shows that the proteins are separated and form different functional groups. To explore what kind of information these groups
represent, we identified the EC number for each enzyme and colored the enzymes in six different groups depending on their top-level EC category. We found that some of the groups that are visually separable represent mainly enzymes within a single EC top-level category. To quantify whether Onto2Vec similarity is representative of EC categorization, we applied $k$-means clustering ($k = 6$) to the protein representations. We evaluated cluster purity with respect to EC top-level classification and found that the purity is 0.42; when grouping enzymes based on their second-level EC classification ($k = 62$), cluster purity increases to 0.60.

### 3.3.3 Ontologies as graphs and axioms

We have developed Onto2Vec, a novel method for learning feature vectors for entities in ontologies. There have been several recent related efforts that use unsupervised learning to generate dense feature vectors for structured and semantically represented data. Notably, there is a large amount of work on knowledge graph embeddings [27, 28, 29, 30], i.e., a set of feature learning methods applicable to nodes in heterogeneous graphs, such as those defined by Linked Data [31]. These methods can be applied to predict new relations between entities in a knowledge graph, perform similarity-based predictions, reason by analogy, or in clustering [32]. However, while some parts of ontologies, such as their underlying taxonomy or partonomy, can naturally be expressed as graphs in which edges represent well-defined axiom patterns [15, 25], it is challenging to represent the full semantic content of ontologies in such a way [26].

It is possible to materialize the implicit, inferred content of formally represented knowledge bases through automated reasoning, and there is a long history in applying machine learning methods to the deductive closure of a formalized knowledge base [33, 34]. Similar approaches have also been applied to knowledge graphs that contain references to classes in ontologies [35]. However, these approaches are still limited to representing only the axioms that have a materialization in a graph-based format.
Onto2Vec is, to the best of my knowledge, the first approach which applies feature learning to arbitrary OWL axioms in biomedical ontologies and includes a way to incorporate an ontology’s deductive closure in the feature learning process. While Onto2Vec can be used to learn feature representations from graph-structures (by representing graph edges as axioms, or triples), the opposite direction is not true; in particular axioms involving complex class expressions, axioms involving disjointness, and axioms involving object property restrictions, are naturally included by Onto2Vec while they are mostly ignored in feature learning methods that rely on graphs alone.

3.3.4 Towards ”trainable” semantic similarity measures

Another related area of research is the use of semantic similarity measures in biology. Onto2Vec generates feature representations of ontology classes, or entities annotated with several ontology classes, and we demonstrate how to use vector similarity as a measure of semantic similarity. In these experiments, we was able to almost match the performance of an established semantic similarity measure [22] when using cosine similarity to compare proteins. It is traditionally challenging to evaluate semantic similarity measures, and their performances differ between biological problems and datasets [20, 108, 109, 110]. The main advantage of Onto2Vec representations is their ability to be used in trainable similarity measures, i.e., problem- and dataset-specific similarity measures generated in a supervised way from the available data. The training overcomes a key limitation in manually created semantic similarity measures: the inability to judge a priori how each class and relation (i.e., axiom) should contribute to determining similarity. For example, for predicting protein-protein interactions, it should be more relevant that two proteins are active in the same (or neighboring) cellular component than that they both have the ability to regulate other proteins. Trainable similarity measures, such as those based on Onto2Vec, can identify the importance of certain classes (and combinations of classes) with regard to a particular
predictive task and therefore improve predictive performance significantly.

Furthermore, Onto2Vec does not only determine how classes, or their combinations, should be weighted in a similarity computation. Semantic similarity measures use an ontology as background knowledge to determine the similarity between two (sets of) classes; how the ontology is used is pre-determined and constitutes the main distinguishing feature among semantic similarity measures [20]. Since Onto2Vec vectors represent both an entity’s annotations and (parts of) the ontology structure, the way in which this structure is used to compute similarity can also be determined in a data-driven way through the use of supervised learning; it may even be different between certain branches of an ontology. We demonstrate that supervised measures outperform binary representations, which shows that combining ontology-based annotations and the ontology structure in a single representation has clear advantages.

### 3.3.5 The informal content of ontologies

In the first part of this chapter, I introduce a new method that combines neural and symbolic methods in biology, and demonstrates significant improvement over state-of-the-art methods. Onto2Vec is specifically targeted at computational biology and the analysis of datasets in which ontologies are used for annotation. While I already demonstrate how Onto2Vec representations can be used to improve predictive models for protein-protein interactions, additional experiments with other ontologies will likely identify more areas of applications.

However, the informative content of biomedical ontologies goes beyond the formal axioms and includes rich natural language text. This informal or meta-data content of ontologies is included by experts in the domain and has the potential to enrich learning from ontologies far beyond "axioms only"-based learning. In the next section of this chapter, I further extend this method to OPA2Vec, the first method to encode for both the formal axioms and the rich meta-data of biomedical ontologies in combination
with the literature to further improve the content of the ontology-based features of biomedical entities. I also show how I can extend the analysis tasks beyond protein interactions prediction to gene–disease association prediction.

3.4 OPA2Vec: Knowledge representation from the formal and informal content of ontologies

In this part of my doctoral work, I propose OPA2Vec, an algorithm that uses asserted and inferred logical axioms in ontologies, combines them with annotation axioms (i.e., meta-data associated with entities or axioms in ontologies) and produces dense vector representations of all entities named in an ontology, or entities associated with classes in an ontology.

Ontologies formalized in the Web Ontology Language (OWL) [114] are based on a Description Logic [113]. In Description Logics, an ontology is described as the combination of a TBox and an ABox [18]. The TBox is a set of axioms that formally characterize classes (e.g., behavior SubClassOf: 'biological process'), while the ABox contains a set of axioms that characterize instances (e.g., SAMN01832237 instanceOf: Biosample). The TBox and ABox together are used by the Onto2Vec method [138] to generate dense vector representations; to achieve this goal, Onto2Vec treats asserted or axioms derived using the reasoner as sentences which form a corpus, and vectors are generated using Word2Vec [124] [125].

In addition to the TBox and ABox (i.e., to the formal, logical axioms characterizing the domain), ontologies contain a large amount of meta-data in the form of annotation axioms [2] [96], and while the axioms are important for automated processing of ontologies, the annotation axioms provide crucial information for humans. OWL annotation axioms relate OWL entities (classes, instances, properties, or axioms) to a literal using an OWL annotation property; I call the literal the “value” of the annotation property [12]. Ontology meta-data consist of the set of non-logical
annotation axioms that describe different aspects of ontology classes, relations, or instances. For example, most ontologies associate entities with a label, a natural language description, several synonyms, etc. While such meta-data are distinct from the formal content of an ontology and therefore not exploited by methods such as Onto2Vec, they nevertheless provide valuable information about ontology classes, relations, and instances.

OPA2Vec (Ontologies Plus Annotations to Vectors) is a novel machine learning method that combines both the formal content of ontologies and the meta-data expressed as OWL annotation axioms to generate feature vectors for any named entity in an ontology; the vectors encode for both the formal and informal content that characterize and constrain the entities in an ontology. OPA2Vec further uses an OWL reasoner, with a choice of either the Elk [123] or HermiT reasoner [122], to access the deductive closure of an ontology. While HermiT supports the complete OWL 2 DL standard [12], its worst case complexity is exponential [18]; Elk only supports the OWL 2 EL subset of OWL 2 [12] but has polynomial complexity and can therefore be applied to larger or more complex ontologies (while losing some of the possible inferences).

The proposed algorithm generates sentences from OWL annotation axioms to form a corpus. From the assertion that an OWL class C has a label L (using the rdfs:label annotation property in the OWL annotation axiom) I generate the sentence C rdfs:label L (using the complete Internationalized Resource Identifier (IRI) for C and rdfs:label), and expressing L as string literal. For example, the relation between class Nuclear periphery (GO:0034399) and its label is expressed as the sentence “<http://purl.obolibrary.org/obo/GO_0034399><http://www.w3.org/2000/01-rdf-schema#label>nuclear periphery”. If C has an annotation axioms relating it to multiple words or sentences, I generate a single sentence in which I ignore sentence or paragraph delimiters. Some annotation prop-
erties do not relate entities to strings, but, for example, to dates, numbers, or other literals. An ontology may contain information about the creation date of a class or axiom; I also generate sentences from these OWL annotation axioms and render the value of the annotation property as a string. For example, the class *Transcription initiation from RNA polymerase I promoter* (GO:0006361) has an annotation axiom that relates it to the date it was created within the GO ontology, and I generate the sentence “<http://purl.obolibrary.org/obo/GO_0006361> <http://www.geneontology.org/formats/oboInOwl# :creation_date> 2011-08-15T03”.

In OPA2Vec, my supervisors and I combine the corpus generated from the metadata (i.e. OWL annotation axioms) and the inferred and asserted logical axioms (using the Onto2Vec algorithm). We then apply a Word2Vec skipgram model on the combined corpus to generate vector representations of all entities in the ontology (for technical details, see Section 4.4).

Natural language words that are used in annotation axioms have a real world linguistic meaning which cannot easily be derived from their use within an ontology alone. Therefore, we use transfer learning in OPA2Vec to assign a semantics to natural language words based on their use in a large corpus of biomedical text. In particular, we pre-train a Word2Vec model on all Medline abstracts, and another model on all open-access fulltext articles available on PubMed Central (PMC), so that natural language words are assigned a semantics (and vector representation) based on their use in biomedical literature (see Section 4.3). The vocabulary in biomedical literature overlaps with the values of annotation properties (e.g., the natural language words used to describe entities in ontologies, or the labels of the entities) but is disjoint with the vocabulary used to refer to the classes, relations, and instances in an ontology (which consists of IRIs). In OPA2Vec, we therefore update the pre-trained Word2Vec model to generate vectors for the entities in the ontology, and we
update the representations of words that overlap between literature and the ontology annotations.

Figure 3.6 illustrates the OPA2Vec algorithm. The input of the algorithm is an ontology $O$ in OWL format as well as a set $A$ of instances and their associations with classes in the ontology. The output of OPA2Vec is a vector representation for each entity in $O$ and $A$ that encodes for the logical axioms and meta-data in $O$ and $A$.

The detailed algorithm of OPA2Vec is described below:

**Algorithm 1: OPA2Vec algorithm for producing vector representations of biological entities**

*Input*: Ontology $O$ defining a set of concepts $c$ (TBox), and set of Entity-Concept associations $c(e)$ (ABox)

*Output*: Vector representation for every entity $e$, $V(e)$

1. Infer the deductive closure of the ontology $O$ using a semantic reasoner (e.g. Elk)
2. Extract annotation properties (meta-data) from ontology $O$
3. Combine formal axioms from 1, ontology meta-data from 2 and Entity-Concept associations $c(E)$ in a text corpus $C$
4. Pre-train Word2Vec on all PubMed abstracts
5. Run Word2Vec model from 4 on corpus $C$
6. Return $V(e)$ for every entity $e$

### 3.4.1 Predicting interactions between proteins

One of the main applications of ontologies is the computation of semantic similarity [20]. As OPA2Vec combines logical axioms and annotation axioms into single vector representations, we expect that we can obtain more accurate feature vectors for biological entities than using the ontology structure alone, and that we can use this to improve the computation of semantic similarity.

To evaluate the hypothesis and demonstrate the potential of using OPA2Vec, we used the Gene Ontology (GO) as a case study (see Section 4.2). We generated a knowledge base using GO, and added either human proteins or yeast proteins as instances. We related each protein to its functions by asserting that a protein $P$ with
Figure 3.6: The detailed workflow of the feature vector generation pipeline of OPA2Vec

function $F$ is an instance of the class has-function some F. We applied OPA2Vec on these two knowledge bases (one including human proteins and the other yeast proteins) and generated vector representations for each protein and ontology class. We then used these vector representations to predict interactions between proteins as characterized in the STRING database [116] by calculating the cosine similarity between each pair of protein vectors and using the obtained value as a prediction score for whether two proteins interact or not. To further improve the prediction performance, We used a neural network model to learn a similarity measure between two feature vectors that is predictive of protein–protein interactions [138]. The steps we followed to predict protein–protein interactions using OPA2Vec are illustrated in Figure 3.7. Figure 3.8 shows the AUC values obtained for OPA2Vec (Figure 3.7).
shows the ROC curves), and the comparison results against Onto2Vec and Resnik’s semantic similarity measure \cite{22} with the Best Match Average strategy \cite{20} for human and yeast.

We found that OPA2Vec significantly improves the performance in predicting interactions between proteins in comparison to both Resnik’s semantic similarity measure and Onto2Vec (e.g., improvement between Onto2Vec and OPA2Vec using cosine similarity).

Figure 3.7: Workflow for protein–protein interaction (PPI) prediction using OPA2Vec.

We found that OPA2Vec significantly improves the performance in predicting interactions between proteins in comparison to both Resnik’s semantic similarity measure and Onto2Vec (e.g., improvement between Onto2Vec and OPA2Vec using cosine
similarity is significant with \( p = 0.031 \) and \( p = 0.041 \) for human and yeast, respectively; one-sided Mann-Whitney U test).

Figure 3.8: AUC values of different methods for PPI prediction for yeast and human. 

- **Onto2Vec** uses formal ontology axioms and compares vectors through cosine similarity;
- **Onto2Vec(NN)** uses a neural network to compare vectors;
- **OPA2Vec-Medline** is our method and uses formal ontology axioms, entity-class associations and annotation properties from the ontology meta-data (labels, description, synonyms, created_by) with a Word2Vec model pre-trained on Medline, and compares vectors through cosine similarity;
- **OPA2Vec-Medline(NN)** is **OPA2Vec-Medline** and uses a neural network to determine similarity between two protein vectors;
- **OPA2Vec-PMC** is similar to **OPA2Vec-Medline** but uses a Word2Vec model pre-trained on fulltext articles in PMC, and compares vectors through cosine similarity;
- **OPA2Vec-PMC(NN)** is **OPA2Vec-PMC** and uses a neural network to determine similarity between two protein vectors;
- **OPA2Vec(No pre-training)** uses same strategy as OPA2Vec but without a pre-trained Word2Vec model;
- **Resnik** is a semantic similarity measure.

To determine the contribution of each annotation property to the performance of OPA2Vec, we restricted the inclusion of annotation properties to each of the following annotation properties which are most frequently used in GO: label (`rdfs:label`), description (`obo:IAO_0000115`), synonym (`oboInOwl:hasExactSynonym`, `oboInOwl:hasRelatedSynonym`, `oboInOwl:hasBroadSynonym`, `oboInOwl:hasNarrowSynonym`), created by (`oboInOwl:created_by`), creation date (`oboInOwl:creation_date`), and OBO-namespace (`oboInOwl:hasOBONamespace`). Table 3.7 show the relative contribution of each of the annotation properties for prediction of protein–protein interac-
tions for human and yeast. We found that the inclusion of the natural language descriptions (obo:IAO_0000115) and the class labels (rdfs:label) results in the highest improvement of performance, while some annotation properties, such as creation date or the namespace, do not improve the prediction. Interestingly, the created_by annotation property adds some minor improvement to the performance. The created_by annotation property is used to keep track of the person who is the creator, or original editor, of a class within an ontology. Classes are often created, edited and defined by experts within a particular domain, and the same expert will add similar or related classes to the GO. Therefore, proteins with associations to classes created by the same person may have higher probability to interact due to having more similar functions.

The analysis shows that annotation properties which describe biological entities in natural language contribute the most to the performance improvements of OPA2Vec. In particular the label and description, synonyms, and created-by properties result in better, more predictive feature vector representations. Therefore, we limited my analysis to the labels, descriptions, synonyms, and creator name from the ontology meta-data in further analysis. The contribution of the major annotation properties in GO are shown in Figure 3.9.

Supervised training can significantly improve the predictive performance when using two vector representations for prediction of biological associations as it has the potential to “learn” custom, task- and dataset-specific similarity measures [138]. Therefore, we trained a deep neural network (see Section 4.6) to predict whether two proteins interact given two protein vector representations as inputs. We found that this supervised approach further improves the performance of OPA2Vec (see Figure 3.8 and 3.10).

Furthermore, we performed all experiments twice, comparing OPA2Vec with the pre-trained models from Medline and PMC. We find that, in general, the model that has been trained on the fulltext articles in PMC performs somewhat better than the
Figure 3.9: ROC curves for PPI prediction comparing the improvement added by each data property in GO meta-data for human and yeast datasets.

Figure 3.10: ROC curves for each prediction method for PPI prediction accuracy for human and yeast.
Table 3.7: Contribution of each annotation property in GO ontology to PPI prediction. I list the AUC values for predicting interacting proteins in human and yeast.

<table>
<thead>
<tr>
<th>Annotation Property</th>
<th>Label</th>
<th>Human</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; <a href="http://purl.obolibrary.org/obo/IAO_0000115">http://purl.obolibrary.org/obo/IAO_0000115</a> &gt;</td>
<td>definition</td>
<td>0.8215</td>
<td>0.8298</td>
</tr>
<tr>
<td>&lt; <a href="http://purl.obolibrary.org/obo/IAO_0100001">http://purl.obolibrary.org/obo/IAO_0100001</a> &gt;</td>
<td>term replaced by</td>
<td>0.7611</td>
<td>0.7703</td>
</tr>
<tr>
<td>&lt; <a href="http://purl.org/dc/elements/1.1/creator">http://purl.org/dc/elements/1.1/creator</a> &gt;</td>
<td>creator</td>
<td>0.7611</td>
<td>0.7701</td>
</tr>
<tr>
<td>&lt; <a href="http://purl.org/dc/elements/1.1/date">http://purl.org/dc/elements/1.1/date</a> &gt;</td>
<td>date</td>
<td>0.7619</td>
<td>0.7693</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#consider">http://www.geneontology.org/formats/oboInOwl#consider</a> &gt;</td>
<td>consider</td>
<td>0.7614</td>
<td>0.7701</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#created_by">http://www.geneontology.org/formats/oboInOwl#created_by</a> &gt;</td>
<td>created_by</td>
<td>0.7697</td>
<td>0.7729</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#creation_date">http://www.geneontology.org/formats/oboInOwl#creation_date</a> &gt;</td>
<td>creation_date</td>
<td>0.7494</td>
<td>0.7536</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasAlternativeId">http://www.geneontology.org/formats/oboInOwl#hasAlternativeId</a> &gt;</td>
<td>has_alternative_id</td>
<td>0.7616</td>
<td>0.7703</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasBroadSynonym">http://www.geneontology.org/formats/oboInOwl#hasBroadSynonym</a> &gt;</td>
<td>has_broad_synonym</td>
<td>0.7683</td>
<td>0.7764</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasDbXref">http://www.geneontology.org/formats/oboInOwl#hasDbXref</a> &gt;</td>
<td>database_cross_reference</td>
<td>0.7617</td>
<td>0.7695</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasExactSynonym">http://www.geneontology.org/formats/oboInOwl#hasExactSynonym</a> &gt;</td>
<td>has_exact_synonym</td>
<td>0.7711</td>
<td>0.7792</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasNarrowSynonym">http://www.geneontology.org/formats/oboInOwl#hasNarrowSynonym</a> &gt;</td>
<td>has_narrow_synonym</td>
<td>0.7731</td>
<td>0.7788</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasOBONamespace">http://www.geneontology.org/formats/oboInOwl#hasOBONamespace</a> &gt;</td>
<td>has_obo_namespace</td>
<td>0.7587</td>
<td>0.7683</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#hasRelatedSynonym">http://www.geneontology.org/formats/oboInOwl#hasRelatedSynonym</a> &gt;</td>
<td>has_related_synonym</td>
<td>0.7680</td>
<td>0.7772</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#id">http://www.geneontology.org/formats/oboInOwl#id</a> &gt;</td>
<td>id</td>
<td>0.7610</td>
<td>0.7700</td>
</tr>
<tr>
<td>&lt; <a href="http://www.geneontology.org/formats/oboInOwl#SynonymTypeProperty">http://www.geneontology.org/formats/oboInOwl#SynonymTypeProperty</a> &gt;</td>
<td>synonym_type_property</td>
<td>0.7609</td>
<td>0.7703</td>
</tr>
<tr>
<td>owl:deprecated</td>
<td></td>
<td>0.7621</td>
<td>0.7709</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>comment</td>
<td>0.7600</td>
<td>0.7692</td>
</tr>
<tr>
<td>rdfs:label</td>
<td>label</td>
<td>0.7973</td>
<td>0.8006</td>
</tr>
<tr>
<td>none</td>
<td>No annotation property</td>
<td>0.7614</td>
<td>0.7701</td>
</tr>
<tr>
<td>All</td>
<td>All annotation properties</td>
<td>0.8792</td>
<td>0.8971</td>
</tr>
</tbody>
</table>

model that has been trained on Medline abstracts alone.

In addition to using ROC curves to evaluate our method in different experiment, we also use precision–recall curves as an additional evaluation metric. Our precision
and recall pairs are calculated for each rank and the resulting values for the area under precision-recall curve (AUPR) are available in Table 3.8 for predicting interacting proteins.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik</td>
<td>0.3422</td>
<td>0.3140</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.2300</td>
<td>0.2741</td>
</tr>
<tr>
<td>OPA2Vec</td>
<td>0.2753</td>
<td>0.2481</td>
</tr>
<tr>
<td>Onto2Vec,NN</td>
<td>0.3375</td>
<td>0.3364</td>
</tr>
<tr>
<td>OPA2Vec,NN</td>
<td><strong>0.3675</strong></td>
<td><strong>0.3476</strong></td>
</tr>
</tbody>
</table>

Table 3.8: Area under PR curve (AUPR) for PPI prediction for human and yeast.

**Protein–protein interaction prediction based on experimental interactions only**

As an additional experiment, we predict protein–protein interactions using STRING’s experimental interactions only (selected by choosing pairs with a confidence score greater than 700). We select positive pairs to be all pairs with confidence score greater than 700 in STRING, while our negatives are sub-sampled from the set of pairs not occurring in STRING in such a way that the cardinality of the positives and negatives is the same. Table 3.9 shows the AUC of the ROC curves resulting from this experiment.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik</td>
<td>0.7521</td>
<td>0.7701</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.7311</td>
<td>0.7474</td>
</tr>
<tr>
<td>OPA2Vec</td>
<td>0.7899</td>
<td>0.7961</td>
</tr>
<tr>
<td>Onto2Vec,NN</td>
<td>0.8104</td>
<td>0.8211</td>
</tr>
<tr>
<td>OPA2Vec,NN</td>
<td><strong>0.8316</strong></td>
<td><strong>0.8523</strong></td>
</tr>
</tbody>
</table>

Table 3.9: AUC values of PPI prediction for human and yeast based on experimental annotations only.
3.4.2 Predicting gene–disease associations

As a second use case to evaluate OPA2Vec and demonstrate its utility, we applied the approach on the PhenomeNET ontology [115] (see Section 4.2). PhenomeNET is a system for prioritizing candidate disease genes based on the phenotype similarity between a disease and a database of genotype–phenotype associations. Phenotypes refer here to concrete developmental, morphological, physiological, or behavioral abnormalities observed in an organism, such as signs and symptoms which make up a disease [139, 140]. PhenomeNET includes the PhenomeNET ontology which integrates several species-specific phenotype ontologies; it can therefore be used to compare, for example, phenotypes observed in mouse models and phenotypes associated with human disease [141]. We used the PhenomeNET ontology and added mouse genes and human diseases to the knowledge base as instances; we then associated each instance with a set of phenotypes. We used the phenotypes associated with unconditional, single gene knockouts (i.e., complete loss of function mutations) available from the MGI database [142] and associated them with their phenotypes, and we used the disease-to-phenotype file from the HPO database [143] to associate diseases from the Online Mendelian Inheritance in Men (OMIM) [144] database to their phenotypes. In total, our knowledge base consists of 18,920 genes and 7,154 OMIM diseases.

We applied the OPA2Vec algorithm to the combined knowledge base to generate vector representations of genes and diseases, and use cosine similarity as well as a neural network to predict gene–disease associations. The corpus generated by OPA2Vec therefore consists of the set of asserted axioms and axioms derived using a semantic reasoner from the PhenomeNET ontology, the set of annotation axioms involving labels, descriptions, synonyms, and creators, and the gene and disease phenotype associations.

We then computed the pairwise cosine similarity between gene vectors and disease
Figure 3.11: AUC values for gene–disease association prediction for different methods, using human gene–disease associations (Human) and identified mouse models of human disease (Mouse) as evaluation sets.

vectors, and we also trained a neural network in a supervised manner to predict gene–disease associations. We evaluated the results using two datasets of gene–disease associations provided by the MGI database, one containing human disease genes and another containing mouse models of human diseases. Figure 3.11 shows the AUC values for gene–disease prediction performance of each approach on the human disease genes and mouse models of human disease (see Figure 3.12 for the ROC curves). The results utilize only labels, descriptions, synonyms, and the created_by annotation properties as they contribute positively to prediction of gene–disease associations for evaluation results using each annotation property. We compared the results to Resnik similarity and Onto2Vec, and found that OPA2Vec outperforms Resnik similarity and Onto2Vec in both evaluation sets. The improvement of OPA2Vec over Onto2Vec using cosine similarity is significant ($p = 0.024$ and $p = 0.026$ for human and mouse; one-sided Mann-Whitney U test), and the improvement of OPA2Vec using cosine similarity over Resnik is significant ($p = 0.0412$ and $p = 0.0307$ for human and mouse; one-sided Mann-Whitney U test). Similar to the results on predicting
interactions based on GO functions associated with gene products, we find that using
the Word2Vec model trained on PMC performs better than the model trained on
Medline abstracts. Similarly to the previous experiment, we also report the are
under precision-recall curve (AUPR) for the prediction of gene–disease associations
for human and mouse in table 3.10.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnik</td>
<td>0.1522</td>
<td>0.1461</td>
</tr>
<tr>
<td>Onto2Vec</td>
<td>0.0929</td>
<td>0.0772</td>
</tr>
<tr>
<td>OPA2Vec</td>
<td>0.0992</td>
<td>0.1214</td>
</tr>
<tr>
<td>OPA2Vec_NN</td>
<td>0.1490</td>
<td>0.1687</td>
</tr>
<tr>
<td>OPA2Vec_NN</td>
<td><strong>0.1724</strong></td>
<td><strong>0.2098</strong></td>
</tr>
</tbody>
</table>

Table 3.10: Area under PR curve (AUPR) for gene–disease association prediction for
human and mouse.

In table 3.11, I report the recall at ranks 1, 10, 20, and 50 using OPA2Vec for
prediction of interacting proteins on human and yeast and prediction of gene–disease
associations human and mouse.
### Table 3.11: Recall at ranks 1, 10, 20, and 50 using OPA2Vec for prediction interacting proteins (on human and yeast) and gene–disease associations (human and mouse).

<table>
<thead>
<tr>
<th></th>
<th>Protein–protein Interaction</th>
<th>Gene–Disease Association</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Human</td>
<td>Yeast</td>
</tr>
<tr>
<td>Rank 1</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Rank 10</td>
<td>0.27</td>
<td>0.33</td>
</tr>
<tr>
<td>Rank 20</td>
<td>0.53</td>
<td>0.47</td>
</tr>
<tr>
<td>Rank 50</td>
<td>0.59</td>
<td>0.56</td>
</tr>
</tbody>
</table>

#### 3.4.3 Discussion of related work

OPA2Vec is a method that combines formal and informal content from biomedical ontologies to produce vector representations of biomedical entities. Several methods are emerging that use different types of semantically represented biological data as well as literature to produce similar kinds of vector representations and use them for prediction tasks [145, 146, 111]. Most of these approaches, including the majority of semantic similarity algorithms [20, 21], are applied to graph-structured data and ignore the axioms that make up many biomedical ontologies. Similarly, ontology-based classification methods that have been developed to predict functions [147] or phenotypes [148] utilize mainly the taxonomy of ontology, ignore other logical axioms as well as all annotation axioms. OPA2Vec is able to utilize the rich meta-data, including textual definitions and labels, that are included in ontologies and in which the scientific community has invested significant resources [96].

There are several text-mining systems that extract or predict gene–disease or gene–phenotype associations from literature and which rely on ontologies as background knowledge [148]. We have compared OPA2Vec to the BeFree text-mining system [130, 131] that also identifies gene–disease associations. We limit the evaluation set to those diseases for which both OPA2Vec and BeFree can make predictions and obtain an AUC of 0.7961 for OPA2Vec and 0.7543 for BeFree, demonstrating that OPA2Vec performs better than BeFree ($p = 0.0365$, one-sided Mann-Whitney U test).
3.4.4 Performance of OPA2Vec using all ontology meta-data annotation properties

As explained in the previous section, we have performed an analysis of the contribution of each meta-data property for the PPI prediction experiment to obtain the properties that contribute most and improve the performance of the prediction. Based on the obtained results, we limited our analysis to include only the properties that have been proven to improve our PPI performance (label, description, synonyms and created-by). However, to have a comprehensive analysis, we decided to also experiment with including all annotation properties for each of the four experiments conducted (PPI prediction on human and yeast and gene-disease association prediction on human and mouse). The obtained results are compared with the original OPA2Vec (trained on a selected subset of annotation properties) for both OPA2Vec and OPA2Vec-NN. The results obtained are shown in Figures 3.13 and 3.14 for PPI prediction and gene-disease prediction respectively. The results obtained from training OPA2Vec on all annotation properties show that there is little to no improvement in the performance when using untrained OPA2Vec, while the performance drops for the trained OPA2Vec when using all annotation properties. This drop in the performance for the neural network trained versions of OPA2Vec is most likely due to the noise introduced by some irrelevant annotation properties in the training dataset.

3.4.5 Potential for discovery of novel disease-associated genes

We apply OPA2Vec to re-analyze data obtained from high-throughput [149] and literature-curated [142] mouse phenotyping experiments in order to discover new mouse models of human disease as well as candidate genes for human genetically based diseases. The main advantage of OPA2Vec in comparing human and mouse phenotypes is the ability to “discover” orthologous phenotypes whereas previously applied methods [149, 41] generally rely on explicitly encoded background knowledge
Figure 3.13: Performance of OPA2Vec using all annotation properties (OPA2Vec + all annotations/ OPA2Vec + all annotations (NN)) compared to the performance of OPA2Vec using a selected subset of properties (OPA2Vec / OPA2Vec (NN)) for PPI prediction for human and yeast.

Figure 3.14: Performance of OPA2Vec using all annotation properties (OPA2Vec + all annotations/ OPA2Vec + all annotations (NN)) compared to the performance of OPA2Vec using a selected subset of properties (OPA2Vec / OPA2Vec (NN)) for gene-disease association prediction for human and mouse.
to determine how phenotypes in mouse and human are related.

We predict candidate genes for over 3,000 orphan diseases in OMIM (all predictions are available from the project website), many of which have not received a prediction previously [149]. We manually analyzed some of the prediction results for candidate genes of orphan disease predicted by OPA2Vec but none of the competing methods. One of my predictions is E2F transcription factor 5 (E2f5, MGI:105091) for an autosomal dominant variant of hydrocephalus (OMIM:123155). The disease is related to a larger deletion on chromosome 8 (8q12.2-q21.2) where it has been hypothesized that a gene associated with an autosomal dominant form of hydrocephalus can be found [150]. Homozygous E2f5 knockout mice develop nonobstructive hydrocephalus [151] as well as several other related abnormalities [152]; the human ortholog of E2f5 in mice is located in the predicted region at 8q21.2, suggesting a possible involvement of E2F5 in hydrocephalus.

Similarly, we predict an involvement of DiGeorge syndrome critical region gene 8 (Dgcr8, MGI:2151114) in Cayler cardiofacial syndrome (OMIM:125520). Cayler cardiofacial syndrome is associated with deletions in 22q11 [153] and associated with abnormalities in facial features and the cardiovascular system [154]. Cardiomyocyte-specific deletion of Dgcr8 in mice leads to left ventricular malfunction, dilated cardiomyopathy, and consequently premature lethality [155], and the human ortholog of Dgcr8 is also located specifically at 22q11.21, i.e., in the region to which Cayler cardiofacial syndrome maps, making Dgcr8 a likely candidate for Cayler cardiofacial syndrome. While determining whether these associations are genuine causal relations will require further functional validation, I believe that the obtained predictions are likely candidates as they are further supported by evidence that is not used in OPA2Vec.
3.4.6 Limitations

The proposed approach has several limitations, some of which I intend to address as future work. While OPA2Vec can utilize OWL axioms for feature learning and prediction, it does not capture information beyond direct associations well; for these purposes, knowledge graph embeddings are more suitable as they can capture information that is more “distant” [32]. One interesting approach in the future may be to combine the axiom- and annotation-based methods such as OPA2Vec with knowledge graph embeddings that also rely on Word2Vec such as those using random walks to explore graphs [35].

A related topic are the representation patterns that associate biological entities with classes in ontologies. If the entities and their relations to ontology classes are included in an OWL ontology, OPA2Vec will generate a representation for them. However, there are multiple design patterns to express particular types of information, including associations between proteins and their functions or genes and their phenotypes [156][157], and OPA2Vec and similar methods will enable their evaluation not only with regard to quality metrics used to evaluate ontologies [158] but also with respect to their potential to be used in machine learning and prediction.

Finally, a major limitation in OPA2Vec is the reliance on Word2Vec which is agnostic to the semantics of operators which have a well-defined meaning in OWL. In the future, I expect to find better approaches to utilize the semantics of operators and quantifiers in OWL ontologies.

3.5 Conclusions

In this chapter, I introduced my two methods Onto2Vec [39] and OPA2Vec [40], the first knowledge representation approaches to learn from all of the formal axioms within ontologies and also, in the case of OPA2Vec, from the rich informal content and annotation properties within ontologies.
3.5.1 Knowledge representation and learning from the logical axioms of ontologies

I first introduce Onto2Vec, a new method that combines neural and symbolic methods in biology, and learns from all the formal axioms within an ontology which is guaranteed to include axioms that cannot easily be represented in the graph structure.

By applying Onto2Vec on the Gene Ontology as a use case, I show how the learned protein vectors can be used to not only predict the existence of interactions between proteins but also predict the exact type of interaction that associates a pair of proteins. I also use visualization to show that Onto2Vec can be used to identify clusters of biomedical entities with shared functional characteristics.

3.5.2 Learning from the formal and informal content of ontologies

In the second part of this chapter, I extend my work to OPA2Vec, a method to produce vector representations for biological entities in ontologies based on the formal logical content in ontologies combined with the meta-data and natural language descriptions of entities in ontologies. I applied OPA2Vec to two ontologies, the GO and PhenomeNET, and I demonstrated that OPA2Vec can significantly improve predictive performance in applications that rely on the computation of semantic similarity. I also evaluated the individual contributions of each ontology annotation property to the performance of OPA2Vec-generated vectors. The results illustrate that the annotation properties that are used to describe details about an ontology class in natural language, in particular the labels and descriptions, contribute most to the feature vectors. I could show that transfer learning, i.e., assigning “meaning” to words by pre-training a Word2Vec model on a large corpus of biomedical literature abstracts, could further significantly improve OPA2Vec performance in the two applications (prediction of protein–protein interactions and prediction of gene–disease
associations. OPA2Vec can comprehensively encode for information in ontologies. The method is based on accepted standards for encoding ontologies, in particular the Web Ontology Language (OWL), and has the potential to include or exclude any kind of annotation property in the generation of its features. OPA2Vec also exploits major developments in the biomedical ontologies community: the use of ontologies as community standards, and inclusion of both human- and machine-readable information in ontologies as standard requirements for publishing ontologies [96, 159]. I therefore believe that OPA2Vec has the potential to become a useful, standard analysis tool in the biomedical domain, supporting any application in which ontologies are being used.

3.5.3 Ontology evaluation using knowledge representation

Ontologies are a constant work in progress. Domain experts regularly update ontology content by including new classes and axioms and by enriching the meta-data and annotation properties associated with the classes. Experts also work on keeping ontologies consistent within the biomedical field by creating links that associate classes from different ontologies [37].

In the next chapter, I show how the knowledge representation methods I proposed here, Onto2Vec and OPA2Vec, can be used as an evaluation tool to assess the quality of the content encoded in some ontologies and these efforts deployed to improve this content. In particular, I use the proposed methods to evaluate the contributions of the axioms and meta-data of the ontology to specific analysis tasks. I also evaluate the quantitative contribution of the axioms that link different ontologies in the biomedical field.
Chapter 4

Task-specific ontology evaluation

4.1 Introduction

In this chapter, I conduct the first quantitative evaluation of the formal and informal content of ontologies for specific analysis tasks. In particular, I show how the two knowledge representation methods that I introduced in the previous chapter, Onto2Vec and OPA2Vec can be used to quantitatively assess the contribution of the axioms and meta-data of different ontologies to the performance of different analysis tasks.

So far in this thesis, I have systematically explored the fundamental importance of ontologies in encoding biomedical knowledge. This importance led to significant efforts to enrich ontologies over the past years by incorporating formalized background knowledge as well as meta-data that improve accessibility and utility of the ontologies [160, 36]. Incorporation of formal axioms contributes to detecting whether ontologies are consistent [161, 162, 163], enables automated reasoning and expressive queries [117, 164, 165], facilitates connecting and integrating ontologies of different domains through the application of ontology design patterns [166, 16], and can be used to guide ontology development [167, 168].

While axioms are mainly exploited through automated tools and methods, ontologies also contain labels, synonyms, and definitions [2]; improving the human-accessible components of ontologies has also been a major focus of ontology development [169]; for example, including “good” natural language definitions and adequate labels is
a requirement for biomedical ontologies in the Open Biomedical Ontologies (OBO) Foundry [160], an initiative to collaboratively develop a set of reference ontologies in the biomedical domains.

The amount of information contained in ontologies, and the rigor with which this information has been created, verified, and represented, may also improve domain-specific data analysis through the provision of background knowledge [170]. Domain-specific background knowledge can limit the solution space in optimization and search problems [170, 171, 172] and therefore allow finding solutions faster or finding better, more generalized solutions. Ontologies and formalized biological knowledge could therefore crucially improve machine learning and applications of Artificial Intelligence in biology.

The Gene Ontology [3] which I used as a major use case all through my doctoral work, holds a particular importance among all ontologies. GO is a biomedical ontology that formally represents several aspects of biological systems, in particular the molecular functions that gene products may have, the biological processes they may be involved in, and the cellular components in which they are located [173]. The GO has been extensively used to provide annotations to gene products through a combination of manual curation of literature and electronic assignments created using algorithms based on sequence similarity, keywords, domain information, and others [174]. Databases such as the GO Annotation (GOA) database [175] use GO to annotate more than 50 million proteins.

Due to its central role and importance in molecular biology, significant resources have been invested in the development of GO. Over the years, substantial efforts have been made to improve the coverage of GO through the addition of new classes [176, 38]. In addition to new classes, GO has also been extended through axioms that characterize the intended meaning of a class formally [36]. Specifically, GO now includes links between GO classes and classes in other biomedical ontologies [37] in
an extended version of GO (which we refer to as “GO-Plus”) [176, 38]. These axioms are particularly useful in keeping GO complete and logically consistent with other ontologies as well as in guiding ontology development [38, 177, 178, 36]. There are now more than 90,000 inter-ontology axioms in GO-Plus that weave GO together with several other ontologies in the biomedical domain.

While these axioms have primarily been developed to tackle the problem of continuously developing GO while maintaining consistency (within GO and other ontologies) as well as to maintain biological accuracy, they also have the potential to significantly improve GO-based data analysis by introducing new associations between classes that are not present in GO but arise through information in other, related ontologies. For example, the GO class Histidine catabolic process to glutamate and formamide (GO:0019556) and the GO class Formamide metabolic process (GO:0043606) are not directly (or closely) related in the GO hierarchy but both are related to the ChEBI class Formamide (CHEBI:16397) through axioms formulated in the Web Ontology Language (OWL) [114], a formal language based on Description Logics [18]. If a data analysis method can utilize the axioms in this formal language, I expect improved performance results when applied to different domains.

A task or method that explicitly relies on the axioms or the meta-data in ontologies can not only be used to improve data analysis but also to evaluate the “quality” of axioms in ontologies in contributing to such an analysis task [11]. Specifically, such a method would enable determining whether axioms and formalized knowledge contribute to biomedical data analysis, and allow evaluating and comparing how much they contribute to particular tasks.

I have previously introduced machine learning methods that make it possible to utilize different components of ontologies – axioms, labels, definitions, and other kinds of meta-data – in machine learning tasks without the need for manual extraction of features (which may introduce a bias). In this chapter, I use the previously dis-
cussed techniques, Onto2Vec [39] and OPA2Vec [40] as well as Node2Vec [179], to predict protein interactions based on functional information and gene–disease associations based on phenotypes. I evaluate the effect of the axioms that have been added to the GO as well as the effect of adding the axioms of additional domain ontologies as the background knowledge. I demonstrate that the formal axioms that have been created for GO and other ontologies improve predictive data analysis by providing background knowledge about biological domains. The proposed approach is also applicable to evaluation of meta-data such as labels and definitions and their contribution to predictive analysis of biomedical data. I find that labels and definitions in ontologies can fill gaps in domain knowledge that are not covered by the axioms and further improve prediction; however, the labels and definitions also have the potential to add noise or bias to prediction results. Finally, through the analysis I also improve the performance of predicting protein interactions and gene–disease associations through ontologies. Overall, the results demonstrate the value that ontologies can provide to biomedical data analysis not merely through their provision of controlled vocabularies but also because they are richly formalized knowledge bases and sources of definitions of domain entities.

4.2 Materials and methods

4.2.1 Ontologies

GO and GO-Plus

Here I use a more recent version of GO [3] in Web Ontology Language (OWL) [114] format from http://www.geneontology.org/ontology/ obtained on April 14, 2018. This version of GO contains 107,762 logical axioms. I also use an updated set of the GO protein annotations from the UniProt-GOA website [http://www.ebi.ac.uk/GOA] on Dec 2, 2018. Similarly to the previous experiments, all associations with
evidence code IEA were filtered, which results in a total of 3,474,539 associations for 749,938 unique proteins.

GO-Plus (downloaded from [http://purl.obolibrary.org/obo/go/extensions/go-plus.owl](http://purl.obolibrary.org/obo/go/extensions/go-plus.owl)) is an extension of GO that contains, in addition to all the logical axioms of GO, additional inter-ontology axioms that describe relations between GO classes and other external biomedical ontologies, in particular: ChEBI (The Chemical Entities of Biological Interest ontology) [180], PO (The Plant Ontology) [181], CL (The Cell Ontology) [182], PATO (Phenotype and Trait Ontology) [183, 140], the Uberon ontology [184], SO (The Sequence Ontology) [185], FAO (Fungal gross anatomy), OBA (Ontology of Biological Attributes), NCBITaxon (NCBI organismal classification), CARO (Common Anatomy Reference Ontology) [186] and PR (Protein Ontology) [187]. I provide below a detailed description for each one of the biomedical ontologies linked to GO in GO-plus:

**The ChEBI Ontology**

I downloaded ChEBI in the OWL format from [http://purl.obolibrary.org/obo/chebi.owl](http://purl.obolibrary.org/obo/chebi.owl) on April 26, 2018. The ChEBI ontology formally describes relations between molecular entities, in particular small chemical compounds [180]. It contains a total of 432,822 logical axioms and 92,015 classes.

**The Plant Ontology (PO)**

I downloaded the OWL version of PO from [http://purl.obolibrary.org/obo/po.owl](http://purl.obolibrary.org/obo/po.owl) on April 26, 2018. This version of PO contains 4,835 axioms and 1,649 classes. PO provides a formal description of the vocabulary related to external and internal plant anatomy and plant development phases. It is mainly used to associate plant structures and development to gene expression and phenotype data [188].
The Cell Type Ontology (CL)

I downloaded CL in OWL from [http://purl.obolibrary.org/obo/cl.owl](http://purl.obolibrary.org/obo/cl.owl) on April 26, 2018. CL contains 17,958 axioms and 3,862 classes. It is an ontology that describes cell types for major animal and plant organisms [182].

Phenotype and Trait Ontology (PATO)

The OWL version of PATO was downloaded from April 26, 2018 from [http://purl.obolibrary.org/obo/pato.owl](http://purl.obolibrary.org/obo/pato.owl). This version contains 5,644 logical axioms and 2,251 different classes. PATO provides a systematic description of phenotypes through the concepts and relationships defined by its axioms [183].

Uberon Ontology

I downloaded the Uberon ontology on April 26, 2018 from [http://purl.obolibrary.org/obo/uberon.owl](http://purl.obolibrary.org/obo/uberon.owl). This OWL version of Uberon contains 65,067 logical axioms and 9,866 classes. Uberon is a multi-species anatomy ontology that describes anatomical structures across multiple species through manually-curated cross-references [184].

Sequence Ontology (SO)

I obtained the SO ontology from [http://purl.obolibrary.org/obo/so.owl](http://purl.obolibrary.org/obo/so.owl) on November 25, 2018. This version of SO contains 5,443 logical axioms and 2,2234 classes. The SO consists of a set of classes and relations that describe the parts of a genomic annotation [185].

Fungal Gross Anatomy Ontology (FAO)

Ontology of Biological Attributes (OBA)

I downloaded the OBA ontology on November 25, 2018 from http://purl.obolibrary.org/obo/oba.owl. This ontology contains 73,377 axioms and 27,365 classes. OBA provides a collection of biological attributes.

NCBI organismal classification (NCBITaxon)

I obtained the NCBITaxon ontology from http://purl.obolibrary.org/obo/ncbitaxon.owl. This OWL version contains 3,653,676 axioms and 1,826,669 classes. This ontology provides a formal classification of different organisms [189].

Common Anatomy Reference Ontology (CARO)

The CARO ontology was obtained on http://purl.obolibrary.org/obo/caro.owl on November 25, 2018. This version contains 209 axioms and 158 classes. The CARO serves as a template to unify the structure of anatomy ontologies [186].

Protein Ontology (PR)

I downloaded the PR ontology from http://purl.obolibrary.org/obo/pro_reasoned.owl on November 4, 2018. This ontology contains 1,312,362 axioms and 400,923 classes. The PR ontology formally represents protein-related entities and their relations at different levels of specificity [187].

Additionally, Table 4.1 summarizes the number of axioms in GO-Plus describing relations to each of these ontologies and shows an example of such axioms for each ontology.

PhenomeNet Ontology

I downloaded the PhenomeNET ontology [41][115] in OWL format from the AberOWL repository http://aber-owl.net [117] on February 21, 2018. PhenomeNET is a
<table>
<thead>
<tr>
<th>Ontology</th>
<th>Number of axioms</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChEBI</td>
<td>69,673</td>
<td>'GDP-L-fucose biosynthetic process' EquivalentTo 'biosynthetic process' and ('has output'(some GDP-L-fucose ))</td>
</tr>
<tr>
<td>PO</td>
<td>935</td>
<td>'metaxylem development' SubClassOf ('results in development of' (some metaxylem ))</td>
</tr>
<tr>
<td>CL</td>
<td>3,859</td>
<td>'epithelial cell differentiation' SubClassOf ('results in acquisition of features of' (some 'epithelial cell' ))</td>
</tr>
<tr>
<td>PATO</td>
<td>205</td>
<td>'supramolecular polymer' SubClassOf ('bearer of' (some polymeric))</td>
</tr>
<tr>
<td>UBERON</td>
<td>17,132</td>
<td>'mammary gland development' SubClassOf ('results in development of' (some 'mammary gland'))</td>
</tr>
<tr>
<td>SO</td>
<td>239</td>
<td>'box C/D snoRNA metabolic process' EquivalentTo ('metabolic process' and has participant (some 'box C/D snoRNA'))</td>
</tr>
<tr>
<td>FAO</td>
<td>99</td>
<td>'cleistothecium development' SubClassOf (results in development of some 'cleistothecium')</td>
</tr>
<tr>
<td>OBA</td>
<td>558</td>
<td>'regulation of post-lysosome vacuole size' SubClassOf (regulates (some 'post-lysosomal vacuole size'))</td>
</tr>
<tr>
<td>CARO</td>
<td>315</td>
<td>'anatomical structure development' EquivalentTo ('Developmental process' and ( results in development of 'anatomical structure'))</td>
</tr>
<tr>
<td>PR</td>
<td>1,914</td>
<td>'tyrosine 3-monooxygenase kinase activity' SubClassOf (has input some ('tyrosine 3-monooxygenase'))</td>
</tr>
<tr>
<td>NCBITaxon</td>
<td>1,136</td>
<td>'chloroplast proton-transporting ATP synthase complex assembly' SubClassOf (only_in_taxon Viridiplantae)</td>
</tr>
</tbody>
</table>

Table 4.1: Number of inter-ontology axioms (with an example) in GO-Plus corresponding to each external ontology.

cross-species phenotype ontology that combines phenotype ontologies, anatomy ontologies, GO, and several other ontologies in a formal manner [41].
4.2.2 Evaluation data sets

Protein-protein interactions (PPI)

To evaluate this work, I predict PPI on three different organisms: human, yeast, and *Arabidopsis thaliana*. The datasets for all three organisms were obtained from the STRING database [116]. The human dataset contains 19,577 proteins and 11,353,057 interactions, the yeast dataset contains 6,392 proteins and 2,007,135 interactions, while the Arabidopsis dataset contains 10,282,070 interactions for 13,261 proteins.

Gene–disease associations

To further evaluate the method, I predict gene–disease associations. The first dataset used in this experiment is the mouse phenotype annotations obtained from the Mouse Genome Informatics (MGI) database [118] on February 21, 2018 with a total of 302,013 unique mouse phenotype annotations. The second dataset used for this experiment is the disease to human phenotype annotations obtained on February 21, 2018 from the Human Phenotype Ontology (HPO) database [119]. I limited my analysis to the OMIM diseases only which resulted in a total of 78,208 unique disease-phenotype associations. To validate the prediction, I also used here the MGI.DO.rpt file from the MGI database to obtain 9,506 mouse gene-OMIM disease associations and 13,854 human gene-OMIM disease associations. To map mouse genes to human genes I used the HMD_HumanPhenotype.rpt file from the MGI database.

4.2.3 Analysis algorithms

The analysis I conduct in this part of my thesis is based on prediction results obtained using embeddings of biological entities (proteins, genes, diseases) obtained from ontologies using two tools: Onto2Vec [39] and OPA2Vec [40]. The obtained embeddings are then trained using a deep multi-layer perceptron neural network to
make predictions.

**Node2Vec**

I used Node2Vec \[179\] to obtain entity embeddings from the biomedical ontologies and their annotations. Node2Vec is a model that learns embeddings of nodes in a graph by applying the Word2Vec model on sequences of nodes. Here, I apply Node2Vec on the ontology graph consisting of subclass, equivalence, and disjointness axioms as well as all types of axioms involving exactly two classes. I use relational patterns \[16\] to map some axioms to graph edges: I identify all axioms of the type \(X\) **SubClassOf**: \(R\) some \(Y\) and \(X\) **EquivalentTo**: \(R\) some \(Y\) (with \(X\), \(Y\), and \(R\) being variables) and for each triple \((X,Y,R)\) I assign an edge labeled \(R\) from \(X\) to \(Y\) (bidirectional in the case of **EquivalentTo**). In addition to these relations, I also represent the annotated entities as a direct edge.

4.3 **Task specific evaluation of ontology content**

4.3.1 **Evaluation of ontology content for functional protein interactions prediction**

We follow a strategy for the external evaluation of ontologies \[11\] and apply the method to the task of predicting interactions between proteins and gene–disease associations. Specifically, we test the impact of ontology axioms and ontology metadata on machine learning applications that rely on ontologies. For this purpose, we use a basic version of the Gene Ontology (GO) \[3\] as the baseline, implement an ontology-based prediction workflow, and evaluate the results. We then compare the performance of ontology-based predictive analysis to the use of GO-Plus in the same workflow and evaluate the results on the same evaluation set. GO-Plus is GO with a large set of formal axioms added that define and constrain GO classes and connect
them to classes that are defined in other ontologies [36]. Furthermore, we add additional background knowledge in the form of the complete set of axioms in biomedical ontologies that are explicitly used in the GO-Plus axioms, and evaluate their impact on predictive performance. Throughout these experiments, we keep training and testing data (comprised of protein identifiers with their associated GO classes) fixed, changing only the amount of background information available through GO.

Since GO-Plus combines all axioms existing in GO with additional axioms that describe relations to other biomedical ontologies, I expect GO-Plus in combination with the axioms and the meta-data of other ontologies to improve predictive performance. I first apply GO and GO-Plus to the task of predicting protein-protein interactions (PPIs), and to account for possible differences between taxa in predicting PPIs, we evaluate the hypothesis on human, yeast, and Arabidopsis proteins and their interactions.

As an illustration, we apply Onto2vec [138] on the axioms of GO-plus to illustrate the classes of each ontology within GO-plus. The obtained embeddings of the classes of each ontology within GO-plus are projected into 2D space using t-SNE [129] and illustrated in Figure 4.1.

To predict PPIs using GO and GO-Plus, we assign GO functions to human, yeast and Arabidopsis proteins based on their annotations in the GOA database [175]. We then apply the Onto2Vec method [39], using either GO or GO-Plus as background knowledge, to obtain ontology embeddings of the proteins. An ontology embedding is a function that maps entities from an ontology (and its annotations) into an n-dimensional vector space [40], and Onto2Vec encodes for ontology-based annotations of entities together with all the axioms in the ontology [39]. I further use the Node2Vec [179] method on the graph structure generated from GO and GO-Plus to generate embeddings (details on how we use Node2Vec to generate the embeddings can be found in Section 4.2.3).
Figure 4.1: t-SNE-based illustration of the vector embeddings of the classes of each ontology in GO-plus.
The workflow generates features (embeddings) for proteins based on the same set of GO annotations but utilizes different sets of axioms (either the axioms in GO, or the extended set of axioms in GO-Plus), and therefore allows us to evaluate the contribution of the ontology axioms to predictions based on these features.

We use the generated features to predict PPIs in two different ways: first, we calculate the cosine similarity between pairs of protein feature vectors (generated through Onto2Vec/Node2Vec), and, second, we train a four-layer fully connected neural network on pairs of vectors, and use a sigmoid output to obtain a prediction confidence score (Onto2Vec-NN/Node2Vec-NN). We evaluate the results of both prediction methods.

Table 4.2 shows the corresponding AUC values for PPI prediction on GO and GO-plus using Onto2Vec and Node2Vec. The ROC curves for PPI prediction for GO and GO-Plus using both Onto2Vec (cosine similarity) and Onto2Vec-NN (neural network) for human, yeast and Arabidopsis thaliana are shown in Figure 4.6. The ROC curves obtained from using Node2Vec for PPI prediction are shown in 4.3.

Table 4.2: AUC values of ROC curves for PPI prediction for GO-Plus and GO using Onto2Vec (cosine similarity) and Onto2Vec-NN (neural network) as well as using Node2Vec (cosine similarity) and Node2Vec-NN (neural network).

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
<th>Arabidopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO_Onto2Vec</td>
<td>0.7660</td>
<td>0.7701</td>
<td>0.7559</td>
</tr>
<tr>
<td>GO_Onto2Vec_NN</td>
<td>0.8779</td>
<td>0.8711</td>
<td>0.8364</td>
</tr>
<tr>
<td>GO_plus_Onto2Vec</td>
<td>0.7880</td>
<td>0.7943</td>
<td>0.7889</td>
</tr>
<tr>
<td>GO_plus_Onto2Vec_NN</td>
<td><strong>0.9021</strong></td>
<td><strong>0.8937</strong></td>
<td><strong>0.8834</strong></td>
</tr>
<tr>
<td>GO_Node2Vec</td>
<td>0.7648</td>
<td>0.7671</td>
<td>0.7601</td>
</tr>
<tr>
<td>GO_Node2Vec_NN</td>
<td>0.8431</td>
<td>0.8568</td>
<td>0.8245</td>
</tr>
<tr>
<td>GO_plus_Node2Vec</td>
<td>0.7713</td>
<td>0.7802</td>
<td>0.7751</td>
</tr>
<tr>
<td>GO_plus_Node2Vec_NN</td>
<td>0.8794</td>
<td>0.8762</td>
<td>0.8573</td>
</tr>
</tbody>
</table>
Figure 4.2: ROC curves for PPI prediction using GO and GO-Plus based on Onto2Vec and Onto2Vec-NN for human, yeast, and Arabidopsis Thaliana.

Figure 4.3: ROC curves for PPI prediction using Node2Vec for GO and GO-Plus for human, yeast, and Arabidopsis Thaliana using cosine similarity and the neural network.
Evaluating the contribution of each set of inter-ontology axioms in GO-plus

The results show that the PPI prediction performance obtained from feature vectors generated using GO-Plus (and the rich set of axioms it contains) outperforms the predictions obtained from using GO axioms alone, both in the unsupervised model (Onto2Vec) and the supervised model (Onto2Vec-NN). The improvement in predictive performance is significant for the Onto2vec prediction based on cosine similarity ($p = 0.021$ for human, $p = 0.034$ for yeast, $p = 0.027$ for Arabidopsis; Mann-Whitney U test), and significant for human and Arabidopsis in the neural network based models ($p = 0.047$ for human, $p = 0.061$ for yeast, $p = 0.039$ for Arabidopsis; Mann-Whitney U test). On the other hand, the improvement in performance using Node2Vec is significant for Arabidopsis using cosine similarity ($p = 0.063$ for human, $p = 0.071$ for yeast, $p = 0.044$ for Arabidopsis; Mann-Whitney U test) and for human and Arabidopsis using the neural network ($p = 0.038$ for human, $p = 0.060$ for yeast, $p = 0.042$ for Arabidopsis; Mann-Whitney U test).

GO-Plus uses axioms from many biomedical ontologies but only includes small parts of these ontologies; I hypothesize that the axioms in the ontologies that are referenced in GO-Plus can contribute additional background knowledge that may further improve data analysis. Therefore, we evaluate the individual contribution of each of the ontologies used in GO-Plus axioms, i.e., we individually evaluate the axioms in the Chemical Entities of Biological Interest (ChEBI) ontology [180], the Plant Ontology (PO) [181], the Cell type Ontology (CL) [182], the Phenotype and Trait Ontology (PATO) [183, 140], the Uberon ontology [184], the Sequence Ontology (SO) [185], the Fungal Gross Anatomy Ontology (FAO), the Ontology of Biological Attributes (OBA), the NCBI organismal classification (NCBITaxon), the Common Anatomy Reference Ontology (CARO) [186] and the Protein Ontology (PR) [187] (a detailed description of each ontology can be found in Section 4.2.1). We perform this
evaluation using Onto2Vec only due to its ability to exploit different types of ontology axioms.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
<th>Arabidopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO (Baseline)</td>
<td>0.7660</td>
<td>0.7701</td>
<td>0.7559</td>
</tr>
<tr>
<td>ChEBI</td>
<td>0.7899(+0)</td>
<td>0.7911(+0)</td>
<td>0.7703(+0)</td>
</tr>
<tr>
<td>PO</td>
<td>0.7752(-0)</td>
<td>0.7761(+0)</td>
<td>0.7671(+0)</td>
</tr>
<tr>
<td>CL</td>
<td>0.7743(+0)</td>
<td>0.7819(+0)</td>
<td>0.7612(+0)</td>
</tr>
<tr>
<td>PATO</td>
<td>0.7657(-0)</td>
<td>0.7707(+0)</td>
<td>0.7563(+0)</td>
</tr>
<tr>
<td>UBERON</td>
<td>0.7780(+0)</td>
<td>0.7824(+0)</td>
<td>0.7645(+0)</td>
</tr>
<tr>
<td>SO</td>
<td>0.7747(+0)</td>
<td>0.7763(+0)</td>
<td>0.7609(+0)</td>
</tr>
<tr>
<td>FAO</td>
<td>0.7660(+0)</td>
<td>0.7712(+0)</td>
<td>0.7544(+0)</td>
</tr>
<tr>
<td>OBA</td>
<td>0.7797(+0)</td>
<td>0.7874(+0)</td>
<td>0.7561(+0)</td>
</tr>
<tr>
<td>CARO</td>
<td>0.7872(+0)</td>
<td>0.7881(+0)</td>
<td>0.7623(+0)</td>
</tr>
<tr>
<td>PR</td>
<td>0.7674(+0)</td>
<td>0.7834(+0)</td>
<td>0.7669(+0)</td>
</tr>
<tr>
<td>NCBI-Taxon</td>
<td>0.7876(+0)</td>
<td>0.7892(+0)</td>
<td>0.7634(+0)</td>
</tr>
<tr>
<td>Average Difference</td>
<td>(+0.0108)</td>
<td>(+0.0114)</td>
<td>(+0.0067)</td>
</tr>
</tbody>
</table>

Table 4.3: AUC values of the ROC curves for PPI prediction showing the contribution of the GO-Plus axioms corresponding to each ontology for human, yeast and Arabidopsis Thaliana. The improvement (blue)/ decrease (red) in performance of each ontology compared to GO is shown between parentheses. The last row shows the average difference of the performance across all ontologies compared to the GO baseline.

We repeat the same workflow as before to generate features: representation of GO annotations of the proteins in human, yeast, and Arabidopsis, and representation
learning with Onto2Vec using GO-Plus as background knowledge; in each experiment we limit the axioms in GO-Plus to those that contain a reference to a particular ontology. We then again apply Onto2Vec to generate features and predict PPIs through cosine similarity or using a neural network (Onto2Vec-NN) on human, yeast and Arabidopsis. The AUC values for the PPI prediction using GO-Plus but limited to the axioms that refer to a particular ontology are shown in Table 4.3. We observe that most of the inter-ontology axioms generally improve the predictive performance, with ChEBI contributing the most to improving PPI prediction and PATO improving the least (even decreasing the performance in some cases). The PO is a plant-specific domain ontology and improves predictive performance mainly when predicting PPIs in Arabidopsis, as can be expected.

**Evaluation of GO-plus in combination with all the axioms of the external ontologies**

As a further experiment, we combine all ontologies, i.e., we add the complete set of axioms from each referenced ontology to the axioms of GO-Plus so that the background knowledge in the referenced ontology becomes available to Onto2Vec as well, and then apply the feature learning and prediction workflow. The detailed AUC values for predicting PPIs based on this comprehensive set of ontologies are shown in 4.4. We observe a similar performance to using only the ontology-specific axioms in GO-Plus.

**Evaluation of the contribution of inter-ontology axioms in GO-plus through the meta-data**

As a final experiment, we replace Onto2Vec with OPA2Vec to evaluate the contribution of ontology meta-data such as labels, synonyms, and definitions, to their predictive performance. We again add each ontology that is referenced in a GO-Plus axiom
Table 4.4: AUC values of the ROC curves for PPI prediction for each external ontology in GO-Plus using Onto2Vec and Onto2Vec-NN. Each prediction method uses all logical axioms from GO, all logical axioms from the referenced ontology, and all GO-Plus axioms describing relations between GO and the given ontology. The improvement (blue)/ decrease (red) in performance of each ontology compared to GO is shown between parentheses. The last row shows the average difference of the performance across all ontologies compared to the GO baseline.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
<th>Arabidopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Onto2Vec</td>
<td>Onto2Vec_NN</td>
<td>Onto2Vec</td>
</tr>
<tr>
<td>GO (Baseline)</td>
<td>0.7660</td>
<td>0.8779</td>
<td>0.7701</td>
</tr>
<tr>
<td>ChEBI</td>
<td>0.7905 (+0.0245)</td>
<td>0.8911 (+0.0132)</td>
<td>0.7920 (+0.0129)</td>
</tr>
<tr>
<td>PO</td>
<td>0.7767 (+0.0007)</td>
<td>0.8790 (+0.0011)</td>
<td>0.7783 (+0.0012)</td>
</tr>
<tr>
<td>CL</td>
<td>0.7804 (+0.0144)</td>
<td>0.8793 (+0.0014)</td>
<td>0.7823 (+0.0122)</td>
</tr>
<tr>
<td>PATO</td>
<td>0.7781 (+0.0121)</td>
<td>0.8788 (+0.0009)</td>
<td>0.7711 (+0.0010)</td>
</tr>
<tr>
<td>UBERON</td>
<td>0.7761 (+0.0101)</td>
<td>0.8795 (+0.0016)</td>
<td>0.7830 (+0.0029)</td>
</tr>
<tr>
<td>SO</td>
<td>0.7890 (+0.0230)</td>
<td>0.8788 (+0.0009)</td>
<td>0.7768 (+0.0067)</td>
</tr>
<tr>
<td>FAO</td>
<td>0.7703 (+0.0043)</td>
<td>0.8781 (+0.0002)</td>
<td>0.7712 (+0.0011)</td>
</tr>
<tr>
<td>OBA</td>
<td>0.7657 (+0.0003)</td>
<td>0.8821 (+0.0042)</td>
<td>0.7874 (+0.0173)</td>
</tr>
<tr>
<td>CARO</td>
<td>0.7742 (+0.0032)</td>
<td>0.8829 (+0.0050)</td>
<td>0.7890 (+0.0189)</td>
</tr>
<tr>
<td>PR</td>
<td>0.7710 (+0.0050)</td>
<td>0.8792 (+0.0013)</td>
<td>0.7859 (+0.0158)</td>
</tr>
<tr>
<td>NCBI Taxon</td>
<td>0.7780 (+0.0120)</td>
<td>0.8857 (+0.0078)</td>
<td>0.7905 (+0.0204)</td>
</tr>
<tr>
<td><strong>Average Difference</strong></td>
<td>(+0.0099)</td>
<td>(+0.0034)</td>
<td>(+0.0122)</td>
</tr>
</tbody>
</table>

to the axioms of GO-Plus, this time also including the meta-data (in the form of annotation axioms) of GO-Plus and the referenced ontology. OPA2Vec (pre-trained on the PubMed corpus) can encode both the axioms and meta-data of ontologies.
and observing the difference to the performance of Onto2Vec can therefore help to evaluate if – and how much – the labels, definitions, and other meta-data contribute.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
<th>Arabidopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OPA2Vec</td>
<td>OPA2Vec_NN</td>
<td>OPA2Vec</td>
</tr>
<tr>
<td>GO (Baseline)</td>
<td>0.8727</td>
<td>0.9033</td>
<td>0.8512</td>
</tr>
<tr>
<td>ChEBI</td>
<td>0.8571 (-0.0156)</td>
<td>0.8801 (+0.0232)</td>
<td>0.8411 0.0101</td>
</tr>
<tr>
<td>PO</td>
<td>0.8680 (-0.0047)</td>
<td>0.8824 (+0.0209)</td>
<td>0.8439 0.0073</td>
</tr>
<tr>
<td>CL</td>
<td>0.8811 (+0.0084)</td>
<td>0.9037 (+0.0004)</td>
<td>0.8561 (+0.0049)</td>
</tr>
<tr>
<td>PATO</td>
<td>0.8562 (-0.0165)</td>
<td>0.8711 (+0.0322)</td>
<td>0.8369 (+0.0143)</td>
</tr>
<tr>
<td>UBERON</td>
<td>0.8714 (-0.013)</td>
<td>0.9033 (+0.000)</td>
<td>0.8514 (+0.0002)</td>
</tr>
<tr>
<td>SO</td>
<td>0.8711 (-0.016)</td>
<td>0.9028 (+0.0005)</td>
<td>0.8509 (+0.0003)</td>
</tr>
<tr>
<td>FAO</td>
<td>0.8709 (-0.018)</td>
<td>0.9011 (+0.0022)</td>
<td>0.8510 (+0.0002)</td>
</tr>
<tr>
<td>OBA</td>
<td>0.8774 (+0.0047)</td>
<td>0.9033 (+0.000)</td>
<td>0.8541 (+0.0029)</td>
</tr>
<tr>
<td>CARO</td>
<td>0.8808 (+0.0081)</td>
<td>0.9037 (+0.0004)</td>
<td>0.8588 (+0.0076)</td>
</tr>
<tr>
<td>PR</td>
<td>0.8829 (+0.0102)</td>
<td>0.9041 (+0.0008)</td>
<td>0.8590 (+0.0078)</td>
</tr>
<tr>
<td>NCBI-Taxon</td>
<td>0.8704 (-0.003)</td>
<td>0.9031 (+0.0002)</td>
<td>0.8508 (+0.0004)</td>
</tr>
<tr>
<td>Average Difference</td>
<td>(-0.0011)</td>
<td>(-0.0070)</td>
<td>(-0.0092)</td>
</tr>
</tbody>
</table>

Table 4.5: AUC values of the ROC curves for PPI prediction for different external ontologies in GO-Plus using OPA2Vec and OPA2Vec-NN. Each prediction method uses the meta-data encoded in GO as well as the meta-data from the external ontologies. In each model, all logical axioms and annotation properties from GO, all logical axioms and all annotation properties from the external ontology, and all GO-Plus inter-ontology axioms are included. The improvement (blue) / decrease (red) in performance of each ontology compared to GO is shown between parentheses. The last row shows the average difference of the performance across all ontologies compared to the GO baseline.
We again predict PPIs in two different ways: calculating the cosine similarity between the obtained protein feature vectors (referred to as OPA2Vec in the results table) and using the feature vectors to train a neural network for PPI prediction (referred to as OPA2Vec-NN in the results tables). The obtained AUC values from this experiment compared to using GO are shown in Table 4.5. We find that the additional meta-data does, in general, not improve predictive performance; on the contrary, the predictive performance drops markedly when adding the meta-data in several ontologies, most notably PATO and ChEBI.

4.3.2 Evaluation of ontology content for gene–disease associations prediction

In the first part of this analysis, we apply GO and GO-Plus to the task of predicting PPIs. Although we utilize PPI datasets from different species for the evaluation in order to generalize the results, it is nevertheless limited to prediction of PPIs and it is unclear if the results also hold for other types of predictive analysis.

We extend the analysis to the evaluation of predicting gene–disease associations based on phenotype similarity [41]. While GO is not a phenotype ontology, it is used in the axioms that make up most phenotype ontologies [140]. We use the cross-species phenotype ontology PhenomeNET [41, 115], which relies on the GO for defining phenotypes, and replace the GO in PhenomeNET with GO-Plus.

We annotate genes with mouse phenotypes from the Mouse Genome Informatics (MGI) [142] database as well as disease phenotypes from the Human Phenotype Ontology (HPO) [143] database, and apply Onto2Vec, Onto2Vec-NN [39], and Node2Vec [179] to encode these phenotypes and the axioms in PhenomeNET as feature vectors (more details on the gene–phenotype and disease–phenotype datasets can be found in Section 4.2.2). We then predict gene–disease associations or mouse models of human disease based on either cosine similarity or a neural network using Onto2Vec,
OPA2Vec, and Node2Vec. We report the results in Table 4.6. The detailed ROC curve figures are shown in Figures 4.4 and 4.5. The ROC curves showing the Node2Vec-based results for gene–disease association prediction are reported in 4.6. The results show that the additional information that GO-Plus provides can significantly improve the overall prediction performance of PhenomeNET in predicting human gene–disease associations and mouse models of human disease ($p = 0.0411$ for mouse and $p = 0.0254$ for human, OPA2Vec, Mann-Whitney U test).

Table 4.6: AUC values of ROC curves for gene–disease prediction using PhenomeNET and when replacing GO in PhenomeNET with GO-Plus as well as using Node2Vec with PhenomeNet and when replacing GO in PhenomeNET with GO-Plus.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phenomenet+GO_Cos</td>
<td>0.7841</td>
<td>0.8431</td>
</tr>
<tr>
<td>Phenomenet+GO_NN</td>
<td>0.8461</td>
<td>0.9141</td>
</tr>
<tr>
<td>Phenomenet+GO-plus_Cos</td>
<td>0.7990</td>
<td>0.8507</td>
</tr>
<tr>
<td>Phenomenet+GO-plus_NN</td>
<td>0.8532</td>
<td>0.9182</td>
</tr>
<tr>
<td>Phenomenet+GO+meta-data_Cos</td>
<td>0.8304</td>
<td>0.8651</td>
</tr>
<tr>
<td>Phenomenet+GO+meta-data_NN</td>
<td>0.8595</td>
<td>0.9188</td>
</tr>
<tr>
<td>Phenomenet+GO-plus+meta-data_Cos</td>
<td>0.8313</td>
<td>0.8672</td>
</tr>
<tr>
<td>Phenomenet+GO-plus+meta-data_NN</td>
<td><strong>0.8761</strong></td>
<td><strong>0.9204</strong></td>
</tr>
<tr>
<td>Phenomenet+GO_Node2Vec</td>
<td>0.7604</td>
<td>0.8104</td>
</tr>
<tr>
<td>Phenomenet+GO_Node2Vec_NN</td>
<td>0.8003</td>
<td>0.8601</td>
</tr>
<tr>
<td>Phenomenet+GO_plus_Node2Vec</td>
<td>0.7794</td>
<td>0.8376</td>
</tr>
<tr>
<td>Phenomenet+GO_plus_Node2Vec_NN</td>
<td>0.8283</td>
<td>0.8882</td>
</tr>
</tbody>
</table>

4.4 Discussion

We developed a method to evaluate the contribution of ontology axioms to computational analysis of biomedical data. We use feature learning methods which are generic and data-driven, and encode for a large set of information contained in ontologies. This choice is motivated by the desire to avoid potential biases, and the ability to use a wide range of formal as well as informal information contained in biomedical ontologies. However, this evaluation is naturally limited to the choice of the embedding methods (Onto2Vec, OPA2Vec, and Node2Vec) as well as the application to the
Figure 4.4: ROC curves for gene–disease prediction comparing PhenomeNET with GO (PhenomeNET + GO) to PhenomeNET with GO-Plus (PhenomeNET + GO-plus) using Onto2Vec with cosine similarity (Cos) and with a neural network (NN) for human gene–disease associations and mouse models of human disease.

Figure 4.5: ROC curves for gene–disease prediction comparing PhenomeNET with GO with the meta-data (PhenomeNET + GO + meta-data) to PhenomeNET with GO-Plus with meta-data (PhenomeNET + GO-plus + meta-data) using OPA2VEC with cosine similarity (Cos) and with a neural network (NN) for human gene–disease associations and mouse models of human disease.
Figure 4.6: ROC curves for gene–disease prediction using Node2Vec comparing PhenomeNET with GO (PhenomeNET + GO) to PhenomeNET with GO-Plus (PhenomeNET + GO-plus) with cosine similarity (Cos) and with a neural network (NN) for human and mouse gene–disease associations.

prediction of PPIs and gene–disease associations and the choice of the Gene Ontology (GO) and the PhenomeNet ontologies as gold standards, and the results may change with different application domains.

Nevertheless, this study allows us to draw several conclusions. First, the results demonstrate that including ontology axioms may add background knowledge that can significantly improve different prediction tasks. Consequently, the results can be used to improve the axioms as well as textual definitions and labels in existing ontologies. For example, we find that the axioms in ChEBI contribute significantly to the prediction of PPIs because ChEBI axioms reveal relations between GO classes that are associated with the same chemical entities but that are not directly related in the GO hierarchy. The axioms also add information in a context-specific manner; for example, axioms from the PO only contribute to predicting protein interactions in Arabidopsis but not other taxa since PO contains plant-specific domain knowledge. Axioms may also add noise to a prediction if they are not well aligned with the prediction task. For example, axioms in the PATO ontology, despite PATO being
significantly smaller in size than ChEBI, do not improve or even decrease performance across several applications.

We also find some evidence that there can be a performance difference when incorporating ontology meta-data into the data analysis. For example, when the OWL annotation axioms of ChEBI are included, the overall PPI prediction performance drops; the labels and definitions in ChEBI often consist of chemical formulas and other properties expressed in symbols or in a mathematical form (e.g., synonyms such as ‘(5Z,8Z,11Z,13E,15R)-15-hydroxyicosa-5,8,11,13-tetraenoic acid’ which are not well represented in literature and therefore not exploited well by my methods. One possibility to overcome this limitation would be combine my embedding method with a chemical named entity recognition and normalization method.

Including the meta-data (labels, definitions, synonyms, etc.) of the PATO ontology in the embeddings consistently decreases predictive performance across all the applications; a possible explanation for this observation is that the labels and definitions in PATO are not well aligned with any of the tasks we intend to perform; my approach provides a quantitative measure that can be used to improve the PATO definitions and labels for the analysis tasks if this is deemed desirable by the PATO developers.

In the next chapter, I attempt to overcome this limitation by proposing a new normalization-based learning method that optimizes learning that combines axioms with natural language text such as the meta-data or the literature for a specific analysis task.

4.5 Conclusions

In this part of my work, I evaluated the contribution of axioms and meta-data in biomedical ontologies towards predictive analysis methods. In my experiments, I do not alter the biological data used for training and evaluation but only alter the back-
ground knowledge encoded in ontologies, using a set of data-driven methods that can encode entities with their ontology-based annotations, together with the ontologies and their axioms, within vector spaces. I find that the background knowledge contained in ontologies has the potential to significantly improve data analysis and machine learning in at least two distinct tasks in bioinformatics: exploiting functional similarity of proteins to predict protein–protein interactions, and exploiting phenotypic similarity between genotypes and diseases to predict gene–disease associations. While this analysis is limited to two tasks, many bioinformatics workflows utilize functional or phenotypic similarity over ontologies and may similarly benefit from exploiting the knowledge contained in richly axiomatized ontologies.

In the next and final section of my doctoral work, I explore approaches to overcome the discussed challenge that causes the ontology meta-data to sometimes have an "adverse effect" on learning from ontologies by introducing noise in the learning process. I tackle this challenge by proposing a new ontology-based normalization of meta-data and the literature in a way that optimizes the integration and learning from both the literature and ontology meta-data.
Chapter 5

Self-normalizing learning on biomedical ontologies

5.1 Introduction

In the previous chapter, through a quantitative evaluation of the content of ontologies, I showed how learning from both the formal content ontologies and natural language data can sometimes be counter-productive and can lead to a drop in the performance compared to learning from axioms only, especially in cases where the content of the ontology and its meta-data is not well aligned with the analysis task of interest.

In this chapter, I propose a new method that overcomes these discussed limitations by optimizing learning that combines the formal content of ontologies with textual data through normalization-based learning.

All through this thesis, I have shown several examples of how ontologies do not serve anymore exclusively as information models that formally represent the massive amounts of biomedical knowledge and its complex relations, but how they have also shown potential in improving machine learning algorithms in several tasks including disease–gene prioritization [190, 191], protein function prediction [72], and protein–protein interaction prediction [192, 138, 193], as well as in natural language processing [194, 195], querying of biomedical data [196], and similar.

Specifically, in biomedical applications, ontologies can contribute to improving machine learning based prediction tasks mainly in two ways. First, they can provide useful features for the biomedical entities of interest (e.g., protein functions using the Gene Ontology (GO) [72], disease phenotype associations based on the Human
Phenotype Ontology (HPO) [197]). Second, they can be used as domain-specific background knowledge that provide annotation property assertions which relate classes to their labels, synonyms, descriptions, etc. In both cases, the predictive performance of the machine learning model can significantly improve if the use of ontologies is combined with existing literature that can complement the information encoded in the ontology annotation assertions [197].

However, in the ontology evaluation analysis conducted and reported in the previous chapter, I have shown that it remains a challenge to combine ontologies and the information contained about the same or related phenomena in the annotation properties and the literature so that the information contained in both can be exploited as background knowledge when building machine learning models. In particular, it is challenging to identify when a class in an ontology has been mentioned in literature and then build a model that can combine information about a class obtained from literature and from the knowledge contained within an ontology.

There are several methods available that use text mining methods to identify the mentions of ontology classes in text [198, 199, 200, 201, 202, 203, 204, 205, 206]. In this last part of my thesis, I propose a method that annotates a literature corpus with classes in an ontology and then normalizes the corpus to the classes within the ontology by replacing each mention of an ontology class in text with the identifiers of the class. I apply this normalization method to biomedical literature as well as to the descriptions of classes within an ontology itself, thereby “self-normalizing” the natural language annotation assertions within an ontology.

I apply these normalization methods so that I can utilize literature and ontologies jointly in machine learning models. Ontology-based machine learning methods mainly use the formal axioms and entity-class annotations without considering natural language information [138, 207, 208]. At the same time, there are a number of methods which apply learning on literature directly to perform biomedical analysis
and prediction tasks [209, 210, 211]. I use the ontology-normalized data set to develop a method that can jointly learn from literature and ontologies through; the proposed method exploits the fact that ontology-based normalization makes the literature and ontologies overlap on the token-level.

I use transfer learning and language models to develop a joint learning method that can generate embeddings for classes in ontologies and entities that are characterized through ontologies. Using a novel deep Siamese neural network architecture, we significantly improve prediction of relations between biological entities based on the embeddings I generate. The results from the experiments show that the combination of ontologies with information in literature and their own annotations, in combination with a deep Siamese neural network, can efficiently characterize semantic similarity and improve the performance in several prediction tasks, including prediction of interactions between proteins, prioritizing gene–disease associations, and prediction the toxicological effects of chemicals.

5.2 Materials and methods

5.2.1 Ontologies

In this part of my thesis, I used the same version of the Gene Ontology (GO) [3] used in the previous chapter and downloaded in Web Ontology Language (OWL) [114] format from http://www.geneontology.org/ontology/ on April 14, 2018. This version of GO contains 107,762 logical axioms. I also downloaded the GO protein annotations from the UniProt-GOA website (http://www.ebi.ac.uk/GOA) on Dec 2, 2018. All associations with evidence code IEA were filtered, which resulted in a total of 3,474,539 associations for 749,938 unique proteins.

As discussed in the previous chapter, GO-Plus (downloaded from http://purl.obolibrary.org/obo/go/extensions/go-plus.owl) is an extension of GO. In addition to all the logical axioms of GO, GO-plus also includes inter-ontology axioms
that link GO classes to other external biomedical ontologies, in particular: ChEBI (The Chemical Entities of Biological Interest ontology) [180], PO (The Plant Ontology) [181], CL (The Cell Ontology) [182], PATO (Phenotype and Trait Ontology) [183, 140], the Uberon ontology [184], SO (The Sequence Ontology) [185], FAO (Fungal gross anatomy), OBA (Ontology of Biological Attributes), NCBITaxon (NCBI organismal classification), CARO (Common Anatomy Reference Ontology) [186] and PR (Protein Ontology) [187].

I downloaded the PhenomeNET ontology [41, 115] in the OWL format from the AberOWL repository http://aber-owl.net [117] on February 21, 2018. PhenomeNET is a cross-species phenotype ontology that combines phenotype ontologies, anatomy ontologies, GO, and several other ontologies in a formal manner [41].

I downloaded the Human Phenotype (HPO) ontology in the OWL format from http://purl.obolibrary.org/obo/hp.owl on February 7, 2019. This ontology contains 34,373 unique classes and 74,426 logical axioms. The HPO ontology formally describes the phenotypic abnormalities frequently encountered in human monogenic diseases [119].

I downloaded the Mammalian Phenotype (MP) ontology in the OWL format from http://purl.obolibrary.org/obo/mp.owl on February 7, 2019. The MP ontology contains 33,356 unique classes and 74,119 logical axioms. The MP ontology classifies and describes phenotypic information related to mice and other mammalian species that are used as models to study human diseases [212, 213].

5.2.2 Normalization methods

Ontology-based dictionaries

To perform the ontology-based normalization of both the PMC articles and the ontology, I extract the labels and synonyms from the annotation property assertions of the ontology to create a lexical dictionary that maps a sequence of words to an
ontology class IRI while filtering the most common words or sequence of words based on their frequency in the British National Corpus. I have created a dictionary based on GO-plus for PPI prediction and a dictionary based on MP, and HPO ontologies for the gene–disease association prediction and the chemical–disease association prediction. For each ontology, the dictionary maps each ontology class (IRI) to the sequence of words that can refer to it, which have been extracted from the labels and synonyms (rdfs:label) and synonyms (hasExactSynonym, hasRelatedSynonym, hasBroadSynonym, hasNarrowSynonym) of each class based on the information available in the ontology annotation property assertions. To allow for morphological variations, I convert all words to their lower case while ignoring white spaces, hyphens and punctuation. If a class is mapped to only one word which happens to be in the list of the most common words in English according to the British National Corpus, this mapping is ignored. The created dictionaries are available at: https://github.com/bio-ontology-research-group/Ontology-based-normalization.

**Whatizit**

The lexical dictionaries created are then used as inputs to Whatizit to annotate PMC articles using biomedical ontology classes. Whatizit is a named entity recognition tool used to annotate text corpora by identifying biomedical classes and terminologies obtained either from known biomedical databases (e.g. Swiss-Prot, DrugBank, etc) or from dictionaries provided by the user. The strength of Whatizit lies in its ability to annotate large files of text relatively fast and in allowing for morphological variations during the normalization. I use Whatizit in this part of my work to annotate all open-access full-text PMC articles obtained from ftp://ftp.ncbi.nlm.nih.gov/pub/pmc on January 10, 2020 using the ontology-based dictionaries. The Whatizit annotation is based on dictionaries that map genes
and disease IDs to their labels in order to annotate PMC articles.

**Self-normalization**

To make better use of the natural language information encoded in the ontology itself, I annotate the natural language definitions of classes with the ontology classes to detect further relations between biomedical classes. In particular, I annotate all values of class definition annotation property assertions in the ontology by using the created ontology-based dictionaries to directly replace all the existing occurrences of labels and synonyms in any description axiom with their corresponding ontology class IDs.

5.2.3 **Embedding methods**

**Text-Based embeddings**

To produce vector representations of biomedical entities based on literature only as a baseline method, I use Whatizit to annotate PMC articles with protein Uniprot IDs for protein interaction prediction, MGI gene IDs and OMIM disease IDs for gene–disease association prediction, and CTD chemical IDs and OMIM disease IDs for chemical–disease association prediction. I then apply the skip-gram model from Word2vec to produce the vector representations of the biological entities based on literature only without including ontology-based information. The obtained vectors are then given as inputs to the neural network in order to predict PPI or gene–disease associations.

**OPA2Vec**

I also use OPA2Vec \[197\] which I previously discussed in Chapter 3 to produce vector embeddings of ontology classes and the entities they annotate. As a recap,
OPA2Vec uses logical axioms as well as annotation property axioms from the ontology, in addition to the known ontology-based associations of biological entities (e.g. protein-GO associations). These annotation axioms use natural language to describe different properties of the ontology classes (labels, descriptions, synonyms, etc.) and they, therefore, form a rich corpus of text for Word2vec. To provide the Word2Vec model with background knowledge on the ontology classes described by the annotation properties, OPA2Vec pre-trains the model on a corpus of biomedical text. Entity-class annotations are also used as an additional source of information to produce the ontology-based embeddings of biological entities.

5.2.4 Siamese neural network

I propose a deep Siamese neural network to predict associations between the embeddings of a pair of biological entities. Siamese neural networks were first proposed in 1994 as an algorithm for signature verification [214]. A Siamese neural network consists of a pair of twin networks in parallel, with similar parameter values, which take distinct samples and are joined by a similarity optimization function [215].

The motivation for using a Siamese neural network comes from the need to overcome data bias and overfitting. While investigating the results previously obtained, it was shown that in some cases predicting associations between two entities using a regular neural network can be misguided by bias in the data. In gene-disease association prediction as an example, some genes which occur more frequently in the training data set tend to be predicted more often than others. In these cases, a regular neural network tends to predict on the evaluation data set based on the gene embedding without necessarily taking the disease embedding into consideration. A Siamese neural network that consists of two parallel neural networks which share the architecture and the weights is forced to learn from both embeddings which overcomes this overfitting issue.
Figure 5.1: Architecture of the deep Siamese neural network we use to predict associations between the embeddings of two biological entities. The Siamese neural network consists of two identical MLP networks that share the same architecture, the same parameters and the same weights.

The Siamese neural network I propose consists of two twin deep feed-forward multi-layer perceptron (MLP) networks with three hidden layers. The MLPs are joined by a contrastive loss function that computes the similarity between the highest-level feature representation of each entity produced by each MLP side. This is accomplished by taking the dot product of the feature representations produced by the twin MLPs followed by a top fully connected layer of 200 neurons and a binary output layer. The parameters of the Siamese neural network are trained using the cross entropy loss and back propagation. Figure 5.1 shows the architecture of the neural network used. To alleviate the issue of over-fitting even further, I use a dropout layer in each MLP as well as an early stopping strategy. To reduce data bias, I over-sample the least frequent diseases in the training data.
5.2.5 Data sets

Functional protein interactions

To evaluate the proposed method, I predict functional protein interactions on three different organisms: human, yeast (S. cerevisiae), and Arabidopsis thaliana. The data sets for all three organisms were obtained from the STRING database [116](http://string-db.org). Similarly to the previous sections, I filtered out interactions with score less than 700 to only keep interactions with strong evidence. The human data set contains 19,577 proteins and 11,353,057 interactions; the yeast data set contains 6,392 proteins and 2,007,135 interactions; and the Arabidopsis data set contains 10,282,070 interactions for 13,261 proteins.

Gene–disease associations

As a second use case, I predict gene–disease associations. The first data set used in this experiment is the mouse phenotype annotations obtained from the Mouse Genome Informatics (MGI) database [118] on February 21, 2018 with a total of 302,013 unique mouse phenotype annotations. The second data set used for this experiment is the disease to human phenotype annotations obtained on February 21, 2018 from the Human Phenotype Ontology (HPO) database [119]. The analysis is limited to the OMIM diseases only which results in a total of 78,208 unique disease-phenotype associations. To validate the prediction, I use the MGI.DO.rpt file from the MGI database to obtain 9,506 mouse gene-OMIM disease associations and 13,854 human gene-OMIM disease associations. To map mouse genes to human genes I use the HMD_HumanPhenotype.rpt file from the MGI database.
Chemical–disease association

To predict associations between chemicals and diseases, I use the Comparative Toxicogenomics Database (CTD) [216] to obtain 108,783 unique chemical–disease associations between 7,248 unique diseases and 10,572 unique chemicals, downloaded on May 5, 2019. Among these chemical–disease associations, 34,573 are therapeutic associations while 62,915 are toxic (marker) associations. To create a balanced data set, I subsample the negative pairs from all the remaining chemical–disease pairs that are not included in the CTD data set. To create the ontology association file, I use GO functional annotations of chemicals and phenotypes annotations for diseases. We obtain a total of 5,416,206 chemical–GO associations from the CTD library, 78,208 disease–phenotype associations from the Human Phenotype Ontology (HPO) database [119], and 124,214 disease–phenotype associations from the disease ontology [134].

5.3 Ontology-based normalization of natural language

Embedding the classes, relations, and instances in ontologies can provide useful features for predictive models that rely on background knowledge, and these embeddings can incorporate ontology axioms as well as natural language annotations such as labels and definitions [207, 217, 218, 138]. However, in the previous chapter, I showed how using the natural language information in ontologies can also add noise, in particular when labels or descriptions use complex terms, such as chemical formulas, which are not easy to recognize in natural language text [219].

In this part of my work, my supervisors and I propose a novel method that more closely integrates ontologies and natural language text, including both literature and the labels, definitions, or synonyms contained within ontologies themselves. We first normalize the natural language text to the ontology using named entity recognition and normalization methods. This recognition and normalization step aims to detect
and then “normalize” terms used to refer to a class in an ontology; here, normalization is the process of ensuring that the symbols used to refer to a class in an ontology in text or within an ontology are identical. To normalize natural language texts, either literature or the descriptions associated with classes in an ontology, we replace every occurrence of a term which refers to an ontology class with the identifier of that class in the ontology (i.e., the class IRI). To recognize mentions of ontology classes in text, we create a dictionary based on the labels and synonyms of the classes in the ontology, and use a dictionary-based named entity recognition and normalization method \cite{203} to replace each mention of an ontology class with the class IRI. We apply this normalization method to any text used as background knowledge. An example is shown in Figure \ref{fig:5.2}.
To learn vector representations of classes and entities, using only the information within ontologies, we modify the OPA2Vec method [197]. OPA2Vec uses a language model applied to a corpus consisting of asserted and deductively inferred axioms, as well as the annotation property assertions, within one ontology; we extend OPA2Vec with our ontology-based normalization method so that all string values used in an annotation property assertion are normalized to the class identifiers within the same ontology.

As I thoroughly discussed in Chapter 3, OPA2Vec has the ability to use transfer learning and thereby incorporate information from literature by pre-training a language model on a literature corpus and then training it further on the corpus generated from the axioms and annotation assertions of the ontology. We apply the same pre-training step in our method but normalize the literature corpus to the ontology before pre-training; we then apply the same transfer learning method as in OPA2Vec.

We expect the normalization of the literature and of ontology class descriptions to improve our ability of the embedding methods to capture previously undetected relations between ontology classes and, therefore, improve their performance in predicting associations between different biomedical entities. This prediction is based on detecting certain similarities within the embedding space, i.e., given two embeddings (such as the embeddings for two proteins, or for a gene and a disease), detect how similar they are (with respect to a certain similarity measure which is generated through supervised learning). We use a deep Siamese neural network architecture [214] that applies two multi-layer perceptron neural networks with shared weights on the two embeddings in a parallel fashion, followed by a dot product, to determine how similar these embeddings are. We use the output of this network as the prediction score for a similarity, or the existence of a relation, between the entities whose embeddings are used as input. The model is trained using a cross-entropy loss. A loss function is applied to merge the parallel networks and calculate the similarity between the
two embeddings to predict the existence (or absence) of an association between the two biological entities. Figure 5.1 provides an overview of the prediction model. A detailed workflow of the proposed method is shown in Figure 5.3.

5.4 Prediction using ontology normalization

5.4.1 Protein interaction prediction

To determine whether our novel method can improve the performance of relation prediction, we apply it to a benchmark data set for predicting interactions between proteins. Predicting interactions between proteins is an established benchmark for testing and comparing similarity-based methods, in particular methods that rely on annotations of proteins with ontologies [126, 138, 197]. We use the GO-plus ontology [176, 38] classes to annotate the literature and its own annotation property assertions (i.e., the natural language definitions of classes), and use the rest of our workflow to produce embeddings of proteins based on GO and the protein-to-GO associations.
The embeddings we generate are then used to predict interactions between proteins for human, yeast and *Arabidopsis Thaliana*, based on the prediction output of our Siamese neural network prediction model. The results from this experiment are referred to as *Annotated OPA2Vec* in Figure 5.4 and Table 5.1. We also evaluate the impact of first normalizing the literature to the ontology (*Annotated PMC*) and the class descriptions within the ontology annotation properties (*Annotated metadata*) separately to assess the contribution of each component in improving the predictive performance of GO-plus for protein interactions. As a baseline method to which we compare our results, we produce vector representations of proteins from PMC articles using Word2vec without including any ontology information (referred to as *PMC only* in the results). We also compare the obtained results to the vector representations obtained from GO-plus using the original OPA2Vec pipeline without any annotation property assertions or text normalization (referred to as *OPA2Vec*).

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Yeast</th>
<th>Arabidopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>PMC only</em></td>
<td>0.8171</td>
<td>0.8529</td>
<td>0.8089</td>
</tr>
<tr>
<td><em>OPA2Vec</em></td>
<td>0.8993</td>
<td>0.8951</td>
<td>0.8582</td>
</tr>
<tr>
<td><em>Annotated_Metadata</em></td>
<td>0.9187</td>
<td>0.9144</td>
<td>0.8726</td>
</tr>
<tr>
<td><em>Annotated_PMC</em></td>
<td>0.9093</td>
<td>0.9065</td>
<td>0.8650</td>
</tr>
<tr>
<td><em>Annotated_OPA2Vec</em></td>
<td><strong>0.9384</strong></td>
<td><strong>0.9256</strong></td>
<td><strong>0.8882</strong></td>
</tr>
</tbody>
</table>

The results show that the annotation of both literature and the class descriptions in the ontology class descriptions performs best among all tested methods. Also, the performance improvement provided by the annotation properties is more significant than that of the literature annotation alone.

5.4.2 Gene–disease association prediction

As a second experiment to evaluate our novel method, we extend our analysis to the task of predicting gene–disease associations based on phenotype similarity using
Figure 5.4: ROC curves for PPI prediction to show the contribution of the ontology-based annotation for literature and ontology descriptions on human, yeast and Arabidopsis.

the PhenomeNet ontology. [41, 115] We use the annotations of human diseases with classes from the Human Phenotype Ontology (HPO) [119], and mouse genes using classes from the Mammalian Phenotype Ontology (MP) [212]. We then follow our prediction workflow and evaluate the performance of text and annotation property normalization using the HPO and MP ontology classes in predicting gene–disease associations for human and mouse, respectively. The results obtained are reported in Figure 5.5 and Table 5.2.

Table 5.2: AUC values for gene–disease association prediction to show the contribution of the ontology-based annotation for literature and ontology descriptions on human and mouse.

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PMC_{only}$</td>
<td>0.8346</td>
<td>0.8695</td>
</tr>
<tr>
<td>$OPA2Vec$</td>
<td>0.8631</td>
<td>0.9286</td>
</tr>
<tr>
<td>Annotated_Metadata</td>
<td>0.8738</td>
<td>0.9368</td>
</tr>
<tr>
<td>Annotated_PMC</td>
<td>0.8657</td>
<td>0.9427</td>
</tr>
<tr>
<td>Annotated_OPA2Vec</td>
<td><strong>0.8999</strong></td>
<td><strong>0.9536</strong></td>
</tr>
</tbody>
</table>

Similar to our evaluation of predicting interactions between proteins, the results from the gene–disease association prediction experiments show that the annotation of the class descriptions of the PhenomeNet ontology and the PMC articles with MP and HPO classes improves the prediction performance for both human and mouse. As our method makes use of information contained in literature, we di-
directly compare our predictive performance to that of BeFree, a text-mining method that predicts gene–disease associations using biomedical named entity recognition \cite{130, 131}. We compare our method with BeFree on the intersection of the BeFree data set and our gene–disease data set consisting of 1,200 human diseases as reported in Table 5.3. Our results show that our normalization based method (Annotated OPA2Vec) significantly outperforms BeFree (p-value of 0.031; Mann-Whitney U test) Figure 5.6 shows the overlapping gene–disease associations between BeFree, Annotated_OPA2Vec, OPA2Vec and the MGI database (ground truth). We find that the Annotated_OPA2Vec shares the highest numbers of positive associations with the MGI database.

**Gene association predictions for orphan diseases**

As part of our validation, we predict gene–disease associations for orphan diseases, i.e., diseases which are suspected or known to have a genetic basis but where no gene association is currently known. Table 5.4 shows predictions for those diseases for which an associated region is known and the gene predicted by us falls precisely
Figure 5.6: Overlapping of positive gene–disease associations between BeFree, Annotated_OPA2Vec, OPA2Vec and the MGI database (ground truth).
Table 5.3: AUC values of gene–disease association prediction on the intersection of our data set and the BeFree data set comparing the performance of BeFree to our methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeFree</td>
<td>0.7550</td>
</tr>
<tr>
<td>PMC_only</td>
<td>0.7258</td>
</tr>
<tr>
<td>Annotated_Metadata</td>
<td>0.8717</td>
</tr>
<tr>
<td>Annotated_PMC</td>
<td>0.8543</td>
</tr>
<tr>
<td>Annotated_OPA2Vec</td>
<td><strong>0.9071</strong></td>
</tr>
</tbody>
</table>

within this region.

5.4.3 Chemical–disease association prediction

In the previous chapter, I have shown how related methods that combine ontology axioms and natural language information in a single machine learning model did not perform well when using information about chemicals, presumably due to the use of chemical formulas in class labels and descriptions [219]. Through the use of ontology-based normalization, we hypothesize that we can somewhat overcome this limitation. One task for which rich information about relevant entities is available through ontologies is the identification of toxicological effects of chemicals, as contained in the Comparative Toxicogenomics Databases (CTD) [216]. We compare the results of using PMC articles only without any ontology-based features, and the use of ontology-based associations with and without text normalization. The results from these experiments are shown in Figure 5.7 and Table 5.5. The results show that the literature and ontology normalization improves our ability to predict chemical–disease associations.

To test whether our method is able to improve over the-state-of-the-art in predicting toxicological effects of chemicals, we compare our best-performing method, Annotated_OPA2Vec, in predicting chemical–disease associations to DigChem, a recent deep learning based method that predicts toxicological effects (chemical–disease
Figure 5.7: ROC curves for chemical–disease association prediction to show the contribution of the ontology-based annotation for literature and ontology descriptions based on the CTD data set.
Figure 5.8: Overlapping positive chemical–disease associations between DigChem, Annotated_OPA2Vec, OPA2Vec and the CTD database (ground truth).

Associations) based on literature [209]. DigChem predictions do not report a confidence score, and therefore our performance comparison is based on accuracy, precision and recall values as reported in Table 5.6. The prediction results show that our normalization based method, Annotated_OPA2Vec, significantly outperforms DigChem (p-value of 0.046; Mann-Whitney U test) Figure 5.8 shows the overlapping chemical–disease associations between DigChem, Annotated_OPA2Vec, OPA2Vec and the CTD database (ground truth). We find that our normalization based method, Annotated_OPA2Vec, shares the highest number of associations with the curated CTD database (ground truth).
To further show the ability of the normalization-based embedding learning for chemicals, we generate embeddings of 1,000 chemicals from the CTD database and use t-SNE [129] to reduce their embeddings to 2D. We color each chemical by its main biological role for six different classes as shown Figure 5.9. This visualization shows that chemicals with similar functions tend to form clusters which suggests that the functional characteristics of the chemicals are mostly conserved in the normalization-based embeddings.

Chemical–disease associations can typically involve two types of relations: ther-
Figure 5.10: ROC curves for predicting toxic and therapeutic associations between chemicals and diseases in the CTD data set.

therapeutic, when the chemical is a drug that can be used to treat the disease, and toxic (marker), when the chemical is a toxin causing the disease. To further analyze the contribution of ontology-based text annotation, we extend our experiments to predicting the specific type of the relation (therapeutic or toxic) that associates the chemical and the disease entity by adapting the architecture of our neural network. The results obtained from this experiments are shown in Figure 5.10 and Table 5.7. We find that in both types of predictions the normalization based learning, Annotated_OPA2Vec, gives the best performance. In general, we can observe that methods using normalization (Annotated_OPA2Vec, Annotated_Metadata, Annotated_PMC) have better prediction performance than methods which do not use normalization (PMC_only, OPA2Vec).

5.5 Discussion

Ontologies have long been used to provide background knowledge. Recently, this background knowledge is increasingly also incorporated into predictive models through semantic similarity or ontology embedding methods. In this chapter, I have demon-
strated that incorporating methods from natural language processing into the feature learning step can further improve the utility of ontologies, mainly through closer alignment between the symbols used in natural language and in formalized theories. While the experiments demonstrate this improvement only in three applications, I believe that it can generalize to other prediction models that rely on combinations of formalized and natural language knowledge.

5.6 Conclusions

In last part of the thesis, I proposed a method that aims to optimize learning using biomedical ontologies and natural language data (literature) using ontology-based normalization. The workflow I propose uses class labels and descriptions from ontologies to normalize both the literature and class descriptions within the ontology metadata. Using a Siamese neural network, the results demonstrated that this method can outperform the-state-of-the-art in several tasks ranging from prediction of interacting proteins through function similarity to prediction of toxicological effects of chemicals. The experiment results showed that the normalization helps prediction methods to learn previously undetected similarities between biomedical classes and entities which improves the learning and predictive performance.
Table 5.4: Gene–disease prediction results for orphan diseases where the predicted gene happens to fall within the known disease segment on the right chromosome.

<table>
<thead>
<tr>
<th>Disease Name</th>
<th>OMIM ID</th>
<th>Gene name</th>
<th>Gene ID</th>
<th>Chromosome</th>
<th>Gene Location</th>
<th>Disease Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facial Cleft</td>
<td>119530</td>
<td>transcription factor AP-2, alpha</td>
<td>104671</td>
<td>chr6</td>
<td>p24.3</td>
<td>p24.3-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>collagen, type XI, alpha 2</td>
<td>88447</td>
<td>chr6</td>
<td>p21.3</td>
<td>p24.3-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vascular endothelial growth factor A</td>
<td>103178</td>
<td>chr6</td>
<td>p21.1</td>
<td>p24.3-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hypocretin (orexin) receptor 2</td>
<td>2680765</td>
<td>chr6</td>
<td>p12.1</td>
<td>p24.3-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dystrobrevin binding protein 1</td>
<td>2137586</td>
<td>chr6</td>
<td>p22.3</td>
<td>p24.3-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>runt related transcription factor 2</td>
<td>99829</td>
<td>chr6</td>
<td>p21.1</td>
<td>p24.3-</td>
</tr>
<tr>
<td>Cleft Palate, Isolated</td>
<td>119540</td>
<td>B9 protein domain 1</td>
<td>1351471</td>
<td>chr17</td>
<td>p11.2</td>
<td>p11.2-</td>
</tr>
<tr>
<td>Cleft Palate, Isolated</td>
<td>119540</td>
<td>tumor necrosis factor receptor superfamily, member 13b</td>
<td>1889411</td>
<td>chr17</td>
<td>p11.2</td>
<td>p11.2-</td>
</tr>
<tr>
<td>Febrile Seizures Familial</td>
<td>121210</td>
<td>EYA transcriptional coactivator and phosphatase 1</td>
<td>109344</td>
<td>chr8</td>
<td>q13.3</td>
<td>q13-q21</td>
</tr>
<tr>
<td>Dyschromatosis Univeralsis</td>
<td>127500</td>
<td>glutamate receptor, metabotropic 1</td>
<td>1351338</td>
<td>chr6</td>
<td>q24.3</td>
<td>q24.2-q25.2</td>
</tr>
<tr>
<td>Preauricular Fistulae</td>
<td>128700</td>
<td>EYA transcriptional coactivator and phosphatase 1</td>
<td>109344</td>
<td>chr8</td>
<td>q13.3</td>
<td>q11.1-q13.3</td>
</tr>
<tr>
<td>Echo Virus II Sensitivity</td>
<td>129150</td>
<td>zinc finger protein 36</td>
<td>99180</td>
<td>chr19</td>
<td>q13.2</td>
<td>q13.1-qter</td>
</tr>
<tr>
<td>Ectrodactyly, Ectodermal Dysplasia and Cleft Lip</td>
<td>129900</td>
<td>ankyrin repeat and SOCS box-containing 4</td>
<td>1929751</td>
<td>chr7</td>
<td>q21.3</td>
<td>q11.2-q21.3</td>
</tr>
<tr>
<td>Ectrodactyly, Ectodermal Dysplasia and Cleft Lip</td>
<td>129900</td>
<td>hepatocyte growth factor</td>
<td>MGI:96079</td>
<td>chr7</td>
<td>q21.11</td>
<td>q11.2-q21.3</td>
</tr>
<tr>
<td>Immune globulin A deficiency I</td>
<td></td>
<td>calcium channel, voltage-dependent, L type, alpha 1S subunit</td>
<td>88294</td>
<td>chr1</td>
<td>q31-q42</td>
<td></td>
</tr>
<tr>
<td>Pseudohypothyroidism</td>
<td></td>
<td>EF-hand domain (C-terminal)</td>
<td>1919127</td>
<td>chr6</td>
<td>p21.3</td>
<td>p24.3-</td>
</tr>
<tr>
<td>Ectrodactyly, Ectodermal Dysplasia and Cleft Lip</td>
<td>129900</td>
<td>EF-hand domain (C-terminal)</td>
<td>129900</td>
<td>chr7</td>
<td>q21.11</td>
<td>q11.2-q21.3</td>
</tr>
<tr>
<td>Malignant Hyperthermia</td>
<td></td>
<td>EF-hand domain (C-terminal)</td>
<td>95410</td>
<td>chr17</td>
<td>q21.2</td>
<td>q11.2-q21.3</td>
</tr>
<tr>
<td>Malignant Hyperthermia</td>
<td></td>
<td>signal transducer and activator of transcription 3</td>
<td>103038</td>
<td>chr17</td>
<td>q21.2</td>
<td>q11.2-q21.3</td>
</tr>
</tbody>
</table>
Table 5.5: AUC values of ROC curves for chemical–disease association prediction to show the contribution of the ontology-based annotation for literature and ontology descriptions based on the CTD data set.

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMC_only</td>
<td>0.7114</td>
</tr>
<tr>
<td>OPA2Vec</td>
<td>0.7741</td>
</tr>
<tr>
<td>Annotated_Metadata</td>
<td>0.8694</td>
</tr>
<tr>
<td>Annotated_PMC</td>
<td>0.8501</td>
</tr>
<tr>
<td>Annotated_OPA2Vec</td>
<td><strong>0.9104</strong></td>
</tr>
</tbody>
</table>

Table 5.6: Performance comparison between the normalization based method and DigChem on chemical–disease association prediction.

<table>
<thead>
<tr>
<th></th>
<th>Annotated_OPA2Vec</th>
<th>DigChem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.873</td>
<td>0.841</td>
</tr>
<tr>
<td>Precision</td>
<td>0.877</td>
<td>0.823</td>
</tr>
<tr>
<td>Recall</td>
<td>0.911</td>
<td>0.880</td>
</tr>
</tbody>
</table>

Table 5.7: AUC values of ROC curves for predicting therapeutic and toxic associations between chemicals and diseases in the CTD data set.

<table>
<thead>
<tr>
<th></th>
<th>Therapeutic</th>
<th>Toxic</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMC_only</td>
<td>0.6787</td>
<td>0.6689</td>
</tr>
<tr>
<td>OPA2Vec</td>
<td>0.7457</td>
<td>0.7551</td>
</tr>
<tr>
<td>Annotated_Metadata</td>
<td>0.8471</td>
<td>0.8519</td>
</tr>
<tr>
<td>Annotated_PMC</td>
<td>0.8336</td>
<td>0.8531</td>
</tr>
<tr>
<td>Annotated_OPA2Vec</td>
<td><strong>0.8594</strong></td>
<td><strong>0.8772</strong></td>
</tr>
</tbody>
</table>
Chapter 6

Conclusion

6.1 Summary

In biology and biomedicine, where a large amount of symbolic structures (ontologies and knowledge graphs) are in use, there are many potential applications for neural-symbolic systems [220]. The current set of methods for knowledge-driven analysis (i.e., analysis methods that specifically incorporate symbolic structures and their semantics) in biology is limited to ontology enrichment analysis [221], applications of semantic similarity [20], and, to a lesser degree, network-based approaches [222].

In my doctoral work, I propose several new methods that solve some of the biological and computation challenges related to the use of biomedical ontologies in computational analysis tasks.

I first propose a new method to predict functions and Gene Ontology classes for non-annotated protein sequences by following a sequence to structure to function paradigm. I show how the proposed method combines local and global structure features with protein interaction networks and functionally discriminative motifs to accurately predict functions for protein sequences. In this work, a key challenge that arises is related to the assessment of the performance protein function prediction which is a special case of the broader challenge of knowledge representation from ontologies.

Then, in what makes the key contribution of my doctoral work, I solve this knowledge representation challenge by proposing a set of machine learning based approaches
that optimize the use of biomedical ontologies as background knowledge. The proposed methods generate ontology-based feature representations of biomedical concepts and entities that can be optimally used by machine learning for a large set of biomedical analysis tasks.

First, with Onto2Vec, I propose a new method in the semantic analysis toolbox, specifically targeted at computational biology and the analysis of data sets in which ontologies are used for annotation. I mainly demonstrate how Onto2Vec representations can be used to improve predictive models for protein-protein interactions based on GO annotations as well be used for clustering and unsupervised classification of enzymes by their functionality.

I then further extend the proposed approach to OPA2Vec, a method to produce vector representations for biological entities in ontologies based on the formal logical content in ontologies combined with the meta-data and natural language descriptions of entities in ontologies. The method is further enhanced by using transfer learning to combine existing biomedical concepts from the literature with the natural language data in ontologies through neural language models. I applied OPA2Vec to two ontologies, the GO and PhenomeNET, and I demonstrated that OPA2Vec can significantly improve predictive performance in applications that rely on the computation of semantic similarity. I also evaluated the individual contributions of each ontology annotation property to the performance of OPA2Vec-generated vectors. OPA2Vec also exploits major developments in the biomedical ontologies community: the use of ontologies as community standards, and inclusion of both human- and machine-readable information in ontologies as standard requirements for publishing ontologies [96, 159].

I then use both of my approaches, Onto2Vec and OPA2Vec, to propose the first quantitative evaluation of the contribution of axioms in biomedical ontologies towards predictive analysis methods. I find that the background knowledge contained in on-
However, my findings show that integrating natural language based data can hurt the performance if the task of interest is not well aligned with the content of the ontology meta-data and literature. To overcome this limitation, I developed a method that optimizes learning from literature and natural language information within biomedical ontologies using named entity recognition and self-normalization. Using a Siamese neural network, I demonstrate that the proposed method can outperform the state-of-the-art in several tasks ranging from prediction of interacting proteins through function similarity to prediction of toxicological effects of chemicals. The experiment results show that the normalization helps prediction methods to learn previously undetected similarities between biomedical classes and entities which improves the learning and predictive performance.

The algorithms I developed and the analyses I conducted in my doctoral work have implications on the further development of knowledge bases and ontologies in the life sciences, in particular as machine learning methods are more frequently applied across the life sciences. The findings motivate the need for further development, and the systematic, application-driven evaluation and improvement, of the content of biomedical ontologies.

I expect that future research on neural-symbolic systems will further extend the results obtained and enable more comprehensive analysis of symbolic representations in biology and biomedicine.

### 6.2 Future Research Work

In this thesis, I propose novel machine learning based methods that optimize the integration of ontologies in biomedical analysis tasks. While the proposed methods have shown successful results in several biomedical applications, there is room for
improvement. In particular, the embedding learning methods lack the ability to encode for the model-theoretic semantics underlying ontologies. To overcome this limitation, I plan on developing a model that can integrate machine learning while also exploiting the semantics underlying ontologies by combining modern machine learning models with approaches based on formal languages and logic, including Markov logic and probabilistic inference.

I also plan to continue working on developing machine learning algorithms that combine both the advantage of formal knowledge and the predictive ability of more complex deep neural networks. Applying graph convolutional neural networks (GCNs) for prediction on interaction networks while using the proposed embeddings as node features is a direction we are currently exploring.

I also aim to diversify the application problems I tend to solve while continuing to work on the biomedical field through applications such as early disease detection as well as learning from Electronic Health Record (EHR) by building a knowledge-base that links clinical observations to basic biology through ontologies and knowledge graphs. This knowledge base can serve as a basis on which I can apply the methods I propose in this thesis to do clinical prediction and analysis tasks such as treatment response prediction.
REFERENCES


discovery from 3,328 gene knockouts by the international mouse phenotyping consortium,” *Nature Genetics*, vol. 49, no. 8, pp. 1231–1238, 8 2017.


with the ontology quality requirements and evaluation method and metrics (oquare),” *PLOS ONE*, vol. 9, no. 8, pp. 1–14, 08 2014. [Online]. Available: [https://doi.org/10.1371/journal.pone.0104463](https://doi.org/10.1371/journal.pone.0104463)


[221] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub,

7 Papers Submitted and Under Preparation