

# **Knowledge Integration and Representation for Biomedical Analysis**

Dissertation  
for the award of the degree

Doctor of Philosophy Ph.D.  
Division of Mathematics and Natural Sciences  
of the Georg-August-Universität Göttingen

within the doctoral Program for Environmental Informatics (PEI)  
of the Georg-August-University School of Science (GAUSS)

submitted by  
Halima Alachram

From  
Lebanon

Göttingen, 2021

Thesis Committee:

Prof. Dr. Edgar Wingender

Department of Bioinformatics,  
University Medical Center Göttingen

Prof. Dr. Winfried Kurth

Department of Ecoinformatics,  
Biometrics & Forest Growth Georg-August  
University of Göttingen

Prof. Dr. Lena Wiese

Institute for Computer Science,  
Department of Mathematics and Computer  
Science  
Goethe University Frankfurt

Members of the Examination Board:

Referee: Prof. Dr. Edgar Wingender

Department of Bioinformatics,  
University Medical Center Göttingen

Co-referee: Prof. Dr. Winfried Kurth

Department of Ecoinformatics,  
Biometrics & Forest Growth Georg-August  
University of Göttingen

Further members of the Examination Board:

Prof. Dr. Lena Wiese

Institute of Computer Science,  
Department of Mathematics and Computer  
Science  
Goethe University Frankfurt

Prof. Dr. Tim Beißbarth

Department of Medical Bioinformatics,  
University Medical Center Göttingen

Prof. Dr. Ulrich Sax

Department of Medical Informatics,  
University Medical Center Göttingen

Prof. Dr. med. Bernd Wollnik

Department of Human Genetics,  
University Medical Center Göttingen

Date of the oral examination: 04.02.2021

## Abstract

Information-based health systems aimed at improving clinical decision-making are appealing as they are able to cope with the rising amount of information that clinicians are experiencing and provide a framework for incorporating validated expertise in health care. Such systems need biomedical analytical expertise, patient-specific data, and a system for reasoning that incorporates data and knowledge to produce and provide clinicians with valuable information during care delivery. Biomedical research has been developed to exploit high-throughput data profiles that provide insights into human disease pathogenesis and diagnosis. The interpretation of high-throughput data involves the comparison of data and knowledge from heterogeneous resources, whether in the biomedical field or in genomics. Enrichment analysis is commonly used for the functional study of gene lists detected by high-throughput techniques like expression microarray experiments. It utilizes statistical methods to detect biological characteristics that are expressed more than expected by chance in a gene set under study. Additionally, healthcare is also seeking closer integration with biomedical data to boost personalized medicine and to provide better treatments. Ontologies, which identify entities and relations used in a domain, play a key role in the automated integration of patient data with relevant knowledge to support clinical research and drug discovery. Moreover, biomedical literature provides valuable insights into the identification of potential treatments, and it can support biomedicine researchers on their way to new findings. With the enormous amount of biomedical literature and the rapid growth of the number of new publications, the wealth of scientific knowledge represented in free text is increasing dramatically. Extracting relevant information and analyzing text data is helpful to discover relationships between biological entities and answer biological questions.

In this thesis, I developed applications that exploit biomedical knowledge represented in different forms and existing in different resources to deliver helpful information in Systems Medicine. The first application is a Java-based enrichment analysis tool which is based on an enrichment function developed in a recent study that uses the logistic regression approach to identify significant categories. I developed a Java command-line interface that uses the logistic regression function in R to integrate the tool into a Java-based platform and to ease its usability by Java users.

Moreover, to facilitate the interoperability between clinical and molecular data existing in biomedical resources, I developed a lexical mapping module in Java to facilitate the mapping of biomedical concepts. I used the module to map the International Classification of Diseases (ICD) terms that represent the names of disease phenotypes in clinical systems to disease concepts in the National Cancer Institute Thesaurus (NCIT) and the Medical Subject Heading (MeSH®) vocabulary. In addition, to deliver the pathway and molecular information integrated into the NCIT ontology, I developed a plugin for the NCIT ontology using the OBA service which is a service that facilitates access to ontologies structures. Using this plugin, I implemented functions that can model disease pathways based on genes.

Furthermore, I used the *word2vec* implementation in two approaches to generate biomedical embeddings. The *word2vec* is one of the most widely used implementations of word embeddings due to its training performance. For the first approach, I used the Dis2Vec model, a vocabulary driven *word2vec* model, to extract disease-drug associations, and I was able to capture visually validated associations. For the second approach, I created and processed a corpus using different preprocessing strategies to obtain embeddings for further comparison. E.g., one passage substituted synonymous terms by their preferred terms in biomedical databases and assigned type labels to words in order to filter similarities for entity types like genes, drugs, or human diseases. To ease the exploration of biomedical concepts and their relations in the embedding, I developed a web service that uses functions to query the embeddings. I validated similarities between entities in obtained embeddings using existing knowledge in biomedical databases. Comparisons showed that relations between entities such as known protein-protein interactions (PPIs), common pathways and cellular functions, or narrower disease ontology groups correlated with higher vector cosine similarity. Word representations as produced by text mining algorithms like *word2vec*, therefore capture biologically meaningful relations between entities. Furthermore, I extracted gene-gene networks from two embedding versions and used them as prior knowledge to train Graph-convolutional neural networks (CNNs) on breast cancer gene expression data to predict the occurrence of metastatic events. Performances of resulting models were compared to Graph-CNNs trained with protein-protein interaction networks or with networks derived using other word embedding algorithms. Graph-CNNs trained with *word2vec*-embedding-derived networks performed best for the metastatic event prediction task compared to PPI or other text mining-based networks.

## Zusammenfassung

Informationsbasierte Gesundheitssysteme zur Verbesserung der klinischen Entscheidungsfindung sind attraktiv, da sie mit der steigenden Menge an Informationen, die Kliniker erfahren, umgehen können und einen Rahmen für die Einbeziehung validierter Fachkenntnisse in die Gesundheitsversorgung bieten. Solche Systeme benötigen biomedizinisches Analysewissen, patientenspezifische Daten und ein Argumentationssystem, das Daten und Wissen enthält, um Kliniker während der Leistungserbringung wertvolle Informationen zu liefern. Die biomedizinische Forschung wurde entwickelt, um Datenprofile mit hohem Durchsatz zu nutzen, die Einblicke in die Pathogenese und Diagnose von Krankheiten beim Menschen liefern. Die Interpretation von Hochdurchsatzdaten beinhaltet den Vergleich von Daten und Wissen aus heterogenen Ressourcen, sei es im biomedizinischen Bereich oder in der Genomik. Die Anreicherungsanalyse wird üblicherweise zur funktionellen Untersuchung von Genlisten verwendet, die mit Hochdurchsatztechniken wie Expressionsmikroarrays nachgewiesen wurden. Es verwendet statistische Methoden, um biologische Eigenschaften nachzuweisen, die in einem untersuchten Gen-Set mehr als zufällig ausgedrückt exprimiert werden. Darüber hinaus strebt das Gesundheitswesen eine engere Integration mit biomedizinischen Daten an, um die personalisierte Medizin zu fördern und bessere Behandlungen anzubieten. Ontologien, die Entitäten und Beziehungen identifizieren, die in einer Domäne verwendet werden, spielen eine Schlüsselrolle bei der automatisierten Integration von Patientendaten mit relevantem Wissen, um die klinische Forschung und die Wirkstoffentdeckung zu unterstützen. Darüber hinaus bietet die biomedizinische Literatur wertvolle Einblicke in die Identifizierung potenzieller Behandlungen und kann biomedizinische Forscher auf ihrem Weg zu neuen Erkenntnissen unterstützen. Mit der enormen Menge an biomedizinischer Literatur und dem rasanten Wachstum der Zahl neuer Veröffentlichungen nimmt der Reichtum an wissenschaftlichen Erkenntnissen im Freitext dramatisch zu. Das Extrahieren relevanter Informationen und das Analysieren von Textdaten ist hilfreich, um Beziehungen zwischen biologischen Einheiten zu entdecken und biologische Fragen zu beantworten.

In dieser Arbeit entwickelte ich Anwendungen, die biomedizinisches Wissen nutzen, das in verschiedenen Formen dargestellt wird und in verschiedenen Ressourcen vorhanden ist, um hilfreiche Informationen in der Systemmedizin zu liefern. Die erste Anwendung ist ein Java-basiertes Tool zur Anreicherungsanalyse, das auf einer Anreicherungsfunktion basiert, die in einer kürzlich durchgeführten Studie entwickelt wurde und den logistischen Regressionsansatz verwendet, um signifikante Kategorien zu identifizieren. Ich habe eine Java-Befehlszeilenschnittstelle entwickelt, die die logistische Regressionsfunktion in R verwendet, um das Tool in eine Java-basierte Plattform zu integrieren und die Benutzerfreundlichkeit für Java-Benutzer zu vereinfachen.

Um die Interoperabilität zwischen klinischen und molekularen Daten in biomedizinischen Ressourcen zu erleichtern, habe ich außerdem ein lexikalisches Mapping-Modul in Java entwickelt, um das

Mapping biomedizinischer Konzepte zu erleichtern. Ich habe das Modul verwendet, um die Begriffe der Internationalen Klassifikation von Krankheiten (ICD), die die Namen von Krankheitsphänotypen in klinischen Systemen darstellen, auf Krankheitskonzepte im National Cancer Institute Thesaurus (NCIT) und im Vokabular des Medical Subject Heading (MeSH®) abzubilden. Um den in die NCIT-Ontologie integrierten Pfad und die molekularen Informationen bereitzustellen, habe ich ein Plugin für die NCIT-Ontologie entwickelt, das den OBA-Dienst verwendet, der den Zugriff auf Ontologiestrukturen erleichtert. Mit diesem Plugin habe ich Funktionen implementiert, die Krankheitswege basierend auf Genen modellieren können.

Darüber hinaus habe ich die Implementierung von word2vec in zwei Ansätzen verwendet, um eine biomedizinische Einbettung zu generieren. Das word2vec ist aufgrund seiner Trainingsleistung eine der am häufigsten verwendeten Implementierungen von Worteinbettungen. Für den ersten Ansatz verwendete ich das Dis2Vec-Modell, ein vokabulargesteuertes word2vec-Modell, um Krankheit-Arzneimittel-Assoziationen zu extrahieren, und konnte visuell validierte Assoziationen erfassen. Für den zweiten Ansatz habe ich einen Korpus mit verschiedenen Vorverarbeitungsstrategien erstellt und verarbeitet, um Einbettungen für den weiteren Vergleich zu erhalten. Beispielsweise ersetzte eine Passage synonym Begriffe durch ihre bevorzugten Begriffe in biomedizinischen Datenbanken und wies Wörtern Typbezeichnungen zu, um Ähnlichkeiten für Entitätstypen wie Gene, Medikamente oder menschliche Krankheiten zu filtern. Um die Erforschung biomedizinischer Konzepte und ihrer Beziehungen in der Einbettung zu vereinfachen, habe ich einen Webdienst entwickelt, der Funktionen zum Abfragen der Einbettung verwendet. Ich habe Ähnlichkeiten zwischen Entitäten in erhaltenen Einbettungen unter Verwendung des vorhandenen Wissens in biomedizinischen Datenbanken validiert. Vergleiche zeigten, dass Beziehungen zwischen Entitäten wie bekannten PPIs, gemeinsamen Pfaden und Zellfunktionen oder engeren Krankheitsontologiegruppen mit einer höheren Ähnlichkeit des Vektorkosinus korrelierten. Wortdarstellungen, wie sie von Text Mining-Algorithmen wie word2vec erzeugt werden, erfassen daher biologisch bedeutsame Beziehungen zwischen Entitäten. Darüber hinaus extrahierte ich Gen-Gen-Netzwerke aus zwei Einbettungsversionen und verwendete sie als Vorwissen, um Graph-Convolutional Neural Networks (CNNs) auf Brustkrebs-Genexpressionsdaten zu trainieren, um das Auftreten metastatischer Ereignisse vorherzusagen. Die Leistungen der resultierenden Modelle wurden mit Graph-CNNs verglichen, die mit Protein-Protein-Interaktionsnetzwerken oder mit Netzwerken trainiert wurden, die unter Verwendung anderer Worteinbettungsalgorithmen abgeleitet wurden. Graph-CNNs, die mit von word2vec-Einbettung abgeleiteten Netzwerken trainiert wurden, zeigten im Vergleich zu PPI oder anderen auf Text Mining basierenden Netzwerken die beste Leistung für die Aufgabe zur Vorhersage metastatischer Ereignisse.

## **Acknowledgments**

It has been a life-changing experience for me to pursue this Ph.D., and it would not have been possible without the support and nurturing of many people who have been instrumental in the successful completion of this work.

First and foremost, I would like to express my sincere appreciation to Prof. Dr. Edgar Wingender for his continued support during the long months I spent undertaking my research work at the Institute of Bioinformatics and at geneXplain GmbH. His profound knowledge helped me to realize the value of critical reasoning. I would also thank him for the opportunities I was given to conduct my research and further my dissertation.

With due respect, I would like to extend my deepest gratitude to Prof. Dr. Andrey Rzhetsky for having me in his lab in Chicago, for his extensive knowledge, and his unwavering support which helped me develop a broader perspective to my thesis

I would also like to thank Prof. Tim for his valuable contribution and directions for completing this thesis. Special thanks go to Philip Stegmaier for his practical suggestions, helpful advice, and insightful comments. My sincere thanks must also go to my thesis committee members: Prof. Dr. Winfried Kurth, Prof. Dr. Lena Wiese, Prof. Dr. Bernd Wollnik and Prof. Dr. Ulrich Sax.

I take immense pleasure to show my greatest appreciation to Prof. Dr. Kifah Tout, who was abundantly helpful and offered invaluable assistance. He has always followed me up and has supported me morally and in every possible way.

My sincere thanks also go to all the department colleagues for their help and support. Special appreciation goes to Gregory for all the insightful discussions. Great thanks to Doris for her wishes and blessings. A special thanks to Maren, Darius, Rayan, Conny, Juergen, Torsten, and many others for their moral and practical support.

Last but not the least, I also want to express my love and heartfelt gratitude to my beloved family and my friends in Lebanon, for their understanding and endless love when it was most required, through my studies.

# Table of Contents

1	Introduction .....	1
1.1.	Thesis Structure.....	5
2	Biological Background .....	7
2.1.	Cellular Organization of Genome .....	7
2.2.	Gene Expression.....	8
2.3.	DNA Microarray .....	9
2.4.	Biological Pathway .....	11
3	Bioinformatics Background.....	14
3.1.	Enrichment Analysis .....	14
3.2.	Network Biology .....	16
3.3.	Biomedical Knowledge Representation and Discovery.....	17
3.3.1.	Ontologies for Biomedical Knowledge Representation .....	17
3.3.2.	Text Mining for Biomedical Knowledge Discovery .....	20
4	Enrichment Analysis Tool.....	23
4.1.	Introduction .....	23
4.2.	Materials and Methods .....	25
4.2.1.	Logistic Regression Approach.....	25
4.2.2.	Gene Ontology .....	25
4.2.3.	Ontology-Based Answers Service (OBA) .....	27
4.2.4.	Reactome.....	27
4.2.5.	Ensembl.....	28
4.2.6.	Biomart .....	28
4.2.7.	Data Sets Processing .....	29
4.2.8.	GO Plugin in OBA.....	30
4.3.	Results .....	32
4.4.	Discussion .....	37
5	Biomedical Knowledge Integration.....	39
5.1.	Introduction .....	39
5.2.	Materials and Methods .....	41
5.2.1.	Terminological Mapping .....	41
5.2.2.	International Classification of Diseases (ICD) .....	46
5.2.3.	Medical Subject Headings (MeSH®) .....	47
5.2.4.	National Cancer Institute Thesaurus (NCIT).....	48



5.2.5.	HumanPSD (Human Proteome Survey Database).....	49
5.3.	Results .....	50
5.3.1.	Lexical Mapping Module Implementation .....	50
5.3.2.	Link Clinical Data to Biomedical Data.....	52
5.4.	Discussion .....	67
6	Biomedical Word Embedding .....	70
6.1.	Introduction .....	70
6.1.1.	Natural Language Processing Techniques and Challenges .....	72
6.2.	Materials and Methods .....	75
6.2.1.	Word Embedding .....	75
6.2.2.	Word2vec .....	76
6.2.3.	Word Embedding Generated using a Preprocessed Text Corpus .....	81
6.2.4.	Biomedical Embeddings Generated from PubMed/MEDLINE® Abstracts .....	82
6.2.5.	Web Service Development .....	87
6.2.6.	Validation of Word Embeddings .....	89
6.2.7.	Examination of Biomedical Embedding Utility .....	90
6.3.	Results .....	97
6.3.1.	Disease-drug Associations .....	97
6.3.2.	Generated Word Embedding.....	102
6.3.3.	Computational Pipeline for Biomedical Embeddings.....	105
6.3.4.	Assessment of our generated Biomedical Word Embedding .....	107
6.3.5.	eBioMeCon: a web service for querying and exploring biomedical concepts .....	113
6.3.6.	Computational Analysis Results .....	124
6.3.7.	Text Corpus Size Effect .....	126
6.3.8.	Graph-CNN Performance Evaluation .....	129
6.3.9.	GLRP to deliver patient-specific subnetworks .....	131
6.4.	Discussion .....	138
7	Conclusion .....	142
7.1.	Summary .....	142
7.2.	Outlooks .....	144
	Bibliography .....	145

## List of Figures

Figure 1. The basic hierarchy within a cell.....	8
Figure 2. Transcription and translation processes.....	9
Figure 3. Process flow for gene expression profiles on the DNA microarray.....	11
Figure 4. WNT signaling pathways control a wide range of developmental.....	13
Figure 5. A simple graph represents an ontology with 3 concepts.....	18
Figure 6. A simple representation of ontology components.....	18
Figure 7. The number of indexed citations that have been added to MEDLINE.....	22
Figure 8. A simple representation of a couple of GO terms.....	26
Figure 9. The mappings of Ensembl gene identifiers and GO/Reactome.....	29
Figure 10. The workflow of extracting and storing the mappings.....	30
Figure 11. The process of generating mappings of Ensembl/Gene ontology identifiers.....	31
Figure 12. Each Ensembl gene ID mapped to a GO term is mapped to all its ancestors.....	32
Figure 13. Command-line interface.....	33
Figure 14. Input file sample.....	34
Figure 15. LRpath Java Tool Architecture.....	35
Figure 16. A results sample of the LRpath Java tool.....	36
Figure 17. A screenshot for the hierarchy of the International Classification of Diseases.....	47
Figure 18. 'Parkinson Disease' term in MeSH® vocabulary.....	48
Figure 19. The NCIT ontology top-level structure in BioPortal.....	49
Figure 20. The preprocessing techniques.....	50
Figure 21. The lexical mapping module architecture.....	52
Figure 22. The “Neoplasms” chapter in the ICD 10 classification hierarchy.....	54
Figure 23. Part of the ICD hierarchy that shows the levels we used.....	55
Figure 24. The “Malignant Breast Neoplasm” class in the NCIT ontology.....	56
Figure 25. An exact lexical mapping that is based on the string matching of an ICD term....	57
Figure 26. Semi-automatic mapping module.....	58
Figure 27. A glimpse of the ICD-NCIT mappings results.....	59
Figure 28. Part of the semantic model in the NCIT.....	59
Figure 29. A screenshot of the disease classes associated with CHEK2 gene in the OBA....	60
Figure 30. The diseases associated with the CHEK2 gene in the console.....	60
Figure 31. The genes associated with Breast neoplasm.....	61
Figure 32. The pathways that the gene PPM1D is element in.....	61
Figure 33. Extracting the disease/pathways associations through genes.....	62
Figure 34. The pathways associated with Breast Carcinoma.....	63
Figure 35. The architecture of the packaged application in Docker.....	63
Figure 36. An ICD term is mapped to the MeSH® term that its superclass is mapped to.....	65
Figure 37. An ICD term matches an entry term (synonym) of a MeSH® term.....	65
Figure 38. A glimpse of the ICD terms at the "low level" mapped to MeSH® terms.....	66
Figure 39. Word2vec converts unique words in a document to distinct real-valued vectors..	76

Figure 40. Visual representation of Euclidean distance (d) and cosine similarity ( $\theta$ ).....	77
Figure 41. Word2vec Architecture. ....	78
Figure 42. The model architectures of CBOW and Skip-gram [172]......	79
Figure 43. Calculating the probability of the output neuron for a word. ....	80
Figure 44. Embedding development workflow. Text processing starts by reading.....	82
Figure 45. Bigram identification example. “Bigram” is a function in Gensim .....	85
Figure 46. Preprocessing procedures. ....	86
Figure 47. Approach workflow: 1. Patients’ microarray data is preprocessed.....	92
Figure 48. The figure depicts a projection of text embedding into three-dimensional space ..	99
Figure 49. Projection of diseases and drugs embeddings of the 'neoplastic process' system ..	99
Figure 50. Projection of diseases and drugs of the 'central nervous' system. ....	100
Figure 51. ‘Zollinger-Ellison syndrome’ and related drugs.....	100
Figure 52. Drugs related to ‘Zollinger-Ellison syndrome’ in the 5-min clinical consult. ....	101
Figure 53. ‘Amphotericin B’ and related diseases.....	101
Figure 54. Diseases related to ‘Amphotericin B’ in the 5-min clinical consult.....	102
Figure 55. The first 10 nearest neighbors of 'wnt4' and 'breast neoplasms'.....	103
Figure 56. Each word in the output vocabulary has a count property of its frequency .....	104
Figure 57. Workflow of the developed computational pipeline. ....	105
Figure 58. Preprocessing workflow. ....	106
Figure 59. Training workflow.....	107
Figure 60. The representation of selected genes, diseases and drugs in the embedding .....	109
Figure 61. The representation of isolated genes with their respective labels. ....	109
Figure 62. The representation of isolated diseases with their respective labels. ....	110
Figure 63. The representation of isolated drugs with their respective labels.....	110
Figure 64. The x-axis consists of the biomedical concepts and y-axis.....	112
Figure 65. eBioMeCon architecture.....	114
Figure 66. A screenshot of the home page of eBioMeCon.....	115
Figure 67. The first 5 nearest neighbors of the 'MDM2' gene. ....	116
Figure 68. The nearest neighbors of the 'MDM2' gene with the API response in JSON .....	117
Figure 69. The similarities between a given word (e.g. TP53 gene) and a list of words.....	118
Figure 70. Similar terms produced by the word analogy "[disease - neoplasms] + drug =?"	119
Figure 71. The first 6 nearest neighbors of the 'TP53' gene with 'gene' output type. ....	120
Figure 72. The combined nearest neighbors of the gene list 'BRCA1, BRCA2, TP53' .....	121
Figure 73. The similarities between entities in a gene list. ....	122
Figure 74. The 'Word Vector' function that returns a vector/vectors of a word/words .....	123
Figure 75. A glimpse of the numerical vector representation.....	123
Figure 76. Validation of the Word2Vec embedding with existing knowledge .....	125
Figure 77. Boxplot of drug- drug cosine similarity distributions with shared genes.....	126
Figure 78. Assessment of similarities between selected terms and their nearest neighbors..	128
Figure 79. PPI subnetworks with the 140 most relevant genes for metastatic patient.....	133
Figure 80. PPI subnetwork with the 140 most relevant genes for metastatic patient .....	134

Figure 81. PPI subnetwork with the 140 most relevant genes for non-metastatic patient.....	135
Figure 82. PPI subnetworks with the 140 most relevant genes for non-metastatic patient ...	136

## List of Tables

Table 1. Embedding results.....	104
Table 2. The results of how weighted underlying networks.....	130
Table 3. Influence of unweighted underlying networks on the performance of Graph CNN	131
Table 4. Four correctly predicted breast cancer patients .....	132

# Acronyms

<b>CNN</b>	Convolutional neural networks
<b>DEG</b>	Differentially Expressed Gene
<b>DNA</b>	Deoxyribonucleic acid
<b>GO</b>	Gene Ontology
<b>GSEA</b>	Gene Set Enrichment Analysis
<b>HPRD</b>	Human Protein Reference Database
<b>ICD</b>	International Classification of Diseases
<b>KEGG</b>	Kyoto Encyclopedia of Genes and Genomes
<b>MeSH®</b>	Medical Subject Headings
<b>NCIT</b>	National Cancer Institute Thesaurus
<b>NLP</b>	Natural language processing
<b>OBA</b>	Ontology-based Answers
<b>PPI</b>	Protein-protein interaction
<b>RNA</b>	Ribonucleic acid
<b>SNP</b>	Single nucleotide polymorphism
<b>UMLS</b>	Unified Medical Language System



# 1 Introduction

Systems biology is rapidly changing the scope of modern healthcare from the diagnosis and treatment of symptom-based diseases to the precision medicine in which patients are the basis of their unique features. Elucidating molecular mechanisms behind diseases is an important field of clinical genomics research, in order to enhance our core understanding of such diseases. This may potentially lead to new targets for diagnosis or treatment. The goal of modern approaches in Systems Medicine is to explore the even more complex interactions of signaling pathways so that individual treatment decisions can be more comprehensive. The need to extend the emphasis on personalized medicine is justified by individualized care decisions and recently invented specialized drugs. Biomedical research seeks to clarify the processes by which a particular disease develops, for which gene expression studies have proven to be a great resource. Gene expression profiling has recently been at the forefront of advancements in personalized medicine, particularly in the cancer domain. The development of high-throughput techniques has allowed scientists to investigate omics data (biological sciences) such as genomes, transcriptomes, proteomes, metabolomes in unprecedented detail [1]. These omics data results in a global health and disease profile and offers new strategies for personalized health surveillance and preventive medicine [1]. Examining the differences between diseased and healthy conditions helps us to understand the disease pathology and to eventually treat it. Detection of differentially expressed genes in disease helps to understand the basic mechanism of disease occurrence.

Moreover, experimental data interpretation generally also demands that clinicians and biologists assess their data to existing knowledge and data sets. A significant trend in biomedical research has recently been the translation of knowledge from basic research into practice. Comprehensive clinical features should be defined in a way that leverages current biomedical knowledge to advance precision medicine. Translational research aims at improving health using a vast variety of biomedical resources, the use of information from experimental data at the diagnosis stage and guiding basic research with patient-finding issues. A crucial challenge for translational medicine informatics is the effective exploitation of multiple types of omics data collected from patient cohorts in studies of human diseases, to develop a more comprehensive picture of the disease, in particular an explanation of how disease mechanisms and disease pathways are linked to changes at the molecular level. Usually,



biomedical knowledge is not very well structured in a unified framework. However, it is dispersed across many biomedical databases as well as scientific literature and might be heterogeneous and complicated. Furthermore, many biomedical systems are not unified a common framework since they have been independently developed, and therefore do not ease navigation across resources. Given these difficulties, there has been recently an evolution towards developing novel approaches for biomedical knowledge representation and developing explicit domain models such as curated and annotated datasets, ontologies, vocabularies, and knowledge bases. An ontology is a systematic knowledge representation within a domain that provides a unified framework of structured concepts and the relationships between them. Ontologies are used to record new information gathered from almost every aspect of today's biomedical research, from conventional biochemical experiments that elucidate particular molecular actors in disease processes to experiments at the omics level that provide systemic tissue-based gene regulation information [2]. Ontological annotations link biological entities to corresponding classes in ontology. Enrichment analysis is a common approach for the use of ontological-based annotations in major knowledge bases of genes and gene products. Using these annotations, enrichment analysis methods determine whether ontology classes have a significant over or under-representation of entities. The efficient use of such annotations involves inferring semantic connections, known as relations, by tracing paths across edges. Biomedical ontologies namely Gene Ontology (GO) [3] are essential methods that use annotation terms for systematic annotation of genes and gene products [2]. An important application of GO is to investigate the functional effect of gene expression in biological and disease processes using gene set enrichment analyses [4]. There are several different methods that use GO annotations in enrichment analyses such as Categorizer [5], GOATOOLS [6], and Map2Slim [7].

Besides, there is an increasing need for the integration and exploitation of heterogeneous biomedical knowledge to support interoperability with healthcare applications for better clinical practice, scientific research, and personalized healthcare. Biomedical ontologies, terminologies, and controlled vocabularies have been commonly used (e.g. MeSH® [8], UMLS [9], etc.), in particular, for the integration of various scientific databases with The International Classification of Diseases (ICD) [10] data. The ICD relies on clinical features that facilitate the understanding of molecular mechanisms of diseases, requiring methods that integrate biomedical data for the classification of diseases in order to meet the needs of precision medicine. As these biomedical resources grow in number and size, redundancy and

inconsistency are increasing between vocabularies. Redundancy occurs when similar terms exist in different vocabularies. However, inconsistency refers to the presence of different terms that represent the same entity in multiple vocabularies. The rapid growth in biological data and information has contributed to understating the utility of ontological approaches in biology and, subsequently, to further efforts to exploit them. One significant potential advantage of these approaches is to bridge the gap between basic biological research and medical applications. The common sense given by ontologies allows the integration of biomedical data formats easier.

In addition to developing biomedical tools for structured data, researchers have been targeting the automated incorporation and use of unstructured data, i.e. biomedical literature, to capture novel findings. Information obtained from semantic-driven literature mining needs to be integrated into established knowledge repositories, thus becoming an integral part of a completely defined and interconnected space for biomedical research knowledge. Many studies have concentrated on extracting knowledge from scientific literature using natural language processing (NLP) methods to promote the discovery and the exploitation of this knowledge [11][12][13][14], which entails large hand-labeled training datasets. One of the most important techniques of NLP is assigning high-dimensional vectors to words, also known as word embeddings, in a text corpus by preserving the syntactic and semantic relationships between words [15]. Many word embedding models and pre-trained word embeddings have been recently published online and applied to several biomedical tasks of NLP [15][16][17]. Wang *et al.* [18] evaluated the performance of word embeddings that were generated using four different corpora namely biomedical literature, clinical notes, Wikipedia, and news articles. Smalheiser *et al.* [19] presented a novel unsupervised method to represent words, text, or phrases as low dimensional vectors based on the word co-occurrence frequency and the similarity between words. Most of the word embeddings are usually trained in the word2vec [15] or GloVe [16] model. These models use information about the co-occurrence of each word to represent it in a distinct vector. Word2vec [15] is one of the most popular word representation implementations, that can capture the meaning of words and similarities between words based on the context.

Knowledge gained from scientific literature can supplement newly obtained experimental data in helping researchers understand the pathological mechanisms underlying diseases. Apart from the semantic incorporation of heterogeneous information sources, the usability of the integrated resource by scientists depends on the availability of knowledge visualization and

exploration tools. Additionally, the integration methods must be modular and must be easy to be used by bench scientists and effective to help them gain new insights from the integrated knowledge bases. The ultimate goal of such interconnected sources of knowledge and exploration tools is to allow scientists to generate novel hypotheses from the knowledge they explore. There are several tools that have been made available and provide the ability to explore the literature for specific information but they are not based on word embedding techniques such as MetaMap [20], MedEvi [21], WhatIzIt [22], Gimli [23], iHOP [24], cTAKES [25], Open Biomedical Annotator [26].

The work of this thesis is part of a consortium project that aims to provide more efficient data use in Systems Medicine by integrating patient clinical and genomics data with pathway knowledge. In particular, the approach is to generate a knowledge base and methods to generate context-specific pathways like patient-specific and disease-specific. The main aim of my thesis work is to tackle the challenge of delivering relevant biomedical knowledge to healthcare applications that help to uncover molecular mechanisms of diseases to promote treatment and drug discovery. This was established by developing applications for biomedical knowledge integration, representation, and exploration. The first application I developed is an enrichment analysis tool that uses the logistic regression method to determine predefined gene sets that are biologically related and enriched with genes differentially expressed. This approach was introduced in a recent study and showed an outperformance by comparing it to other enrichment analysis methods. I developed a Java command interface that integrates the logistic regression function that was originally implemented in R and uses GO and Reactome categories annotated with ENSEMBL gene identifiers as predefined data sets to test significant genes. The tool is a standalone Java application that was developed to be integrated into a Java platform and be available to be used by researchers and clinicians. It was tested with a data sample and the analysis was successfully performed.

Furthermore, in order to promote the comparison of clinical and biological data to reference knowledge bases and to already existing data sets, I mapped ICD codes, used in clinical systems to define disease phenotypes, to disease concepts in the NCIT and the MeSH® vocabulary. I developed a lexical mapping module to establish the mapping between concepts in different resources and by using other mapping strategies. The lexical mapping approach is based on matching two concepts lexically using string similarity metrics. To provide molecular and pathway information related to diseases, I used the NCIT ontology structure to develop functions that help to model disease pathways. The functions were implemented into a plugin

in the Ontology-Based Answers (OBA) [27] service that provides access to ontology structures using specific functions for specific ontologies.

Further, extracting relevant information and analyzing text data is helpful to discover relationships between biological entities and answer biological questions. Making use of the *word2vec* approach, I generated word vector representations based on a corpus consisting of over 16 million PubMed abstracts. Preprocessing techniques were applied to generate embedding similarities for comparison purposes. I annotated a number of biological entities to get more insights into the embedding information and to facilitate the extraction of entities of a particular type. To ease the processing of other text corpora and the development of word embedding, I developed a pipeline based on the implemented methods. Additionally, I developed a web service that provides both a graphical web interface as well as a RESTful API to explore the resulting embedding. To derive biological interpretations and explain the variation of the similarities between entities, I performed computational analyses using existing knowledge in biomedical databases. The analysis results showed that relations between entities such as known PPIs, common pathways and cellular functions, or narrower disease ontology groups correlated with higher vector cosine similarity. In addition, I assessed the effect of corpus size on the variability of word representations. Moreover, created a gene-gene network and used it as prior knowledge to structure gene expression data of breast cancer patients in order to predict the occurrence of metastatic events. Graph-CNNs trained with *word2vec*-embedding-derived networks performed best for the metastatic event prediction task compared to PPI or other text mining-based networks.

## **1.1. Thesis Structure**

This thesis is organized as follows. In chapter 2, I introduce the biological facts and techniques that help to understand the molecular mechanisms of diseases by presenting basic biological information related to the subjects detailed in the following chapters. In chapter 3, I present the bioinformatics techniques that can be used to interpret the functions of genes that play a role in disease development. Further, I introduce the biomedical resources used to represent biomedical knowledge and are essential keys in biomedical translational research. Moreover, I hereby present the role of literature information in knowledge representation and discovery. To present the main topics of my work, each of the following chapters is structured into four sections: a short introduction describes the specific problems addressed in the chapter, the

specific materials and methods used to solve these problems, the results obtained and a discussion. In chapter 4, I present the enrichment analysis tool I developed that uses the logistic regression approach. The chapter starts by describing the study approach that introduced the logistic regression function, followed by describing the Java-based tool I developed. Chapter 5 depicts the need for biomedical knowledge integration to bridge the gap between clinical systems and existing biomedical knowledge resources. This is followed by presenting the lexical mapping approach I used to map The ICD terms to disease concepts in the NCIT and the MeSH® vocabulary accompanying other strategies. Further, I present the functions I developed to model disease pathways. In chapter 6, I describe the process of developing word embedding from biomedical literature. Besides, I present the pipeline I developed to process text corpus and to generate word embedding, and the web service that aims to ease the exploration of biomedical concepts in the embedding. Eventually, the results of this chapter comprise the statistical analyses I performed to evaluate the biological meanings of the similarities between entities in the embedding. In addition, the results of training graph CNN using the gene-gene embedding are presented to evaluate the biological utility of the embedding. Finally, concluding remarks and future works are presented in chapter 7.

## **2 Biological Background**

### **2.1 Cellular Organization of Genome**

Cells are the essential components of any living thing. Bodies are made up of trillions of cells that provide the body with a structure, absorb nutrients from food to convert them into energy, and perform specialized functions. Every cell comprises various organelles, all of which have a significant role as a part of the cell cycle, such as waste decomposition or energy production. The cell nucleus is the most important organelle that houses the cell's hereditary material, or DNA (deoxyribonucleic acid), and coordinates its growth and reproduction. In humans and nearly every other organism, DNA is the hereditary substance. The complete set of the DNA in each cell is called its genome. DNA has a double helix structure, that is, two long strands appear twisted around each other (Figure 1). Each of the two strands is made up of a sequence of entities called nucleotides. Each of these nucleotides is made of a phosphate molecule, a nitrogen base, and a sugar molecule. The nitrogen bases are of four types: adenine (A), guanine (G), thymine (T), and cytosine (C) (Figure 1). The two strands of the DNA molecule are joined by hydrogen links between the bases, with a base pair formed by adenine with thymine, and another base pair formed by cytosine with guanine. The orders of these four bases along a strand determine the genetic code which is the biological instructions. Human DNA contains around 3 billion bases and over 99% of these bases are shared across all humans. The DNA of almost each human body cell exhibits the same nucleotide sequence. DNA is present in all the body cells, except those that do not have a nucleus, such as mature red blood cells or cornified nail and skin cells. DNA does more than determining the structure and characteristics of living things, it is also the hereditary material that is passed to the next generation in organisms of all types.

The DNA molecule is packed into threaded structures called "chromosomes" inside the nucleus of every cell. Every chromosome consists of DNA wrapped up around proteins known as histones that maintain their structure (Figure 1). The number of chromosomes is constant in each cell in the body (except sex cells which only have half sets) and constant for all members of a species. Every human cell consists of 46 chromosomes and each of them contains highly condensed and coiled DNA comprising millions of gene sequences. Each cell nucleus contains  $3 \times 10^9$  base pairs of the DNA distributed over 23 chromosome pairs. DNA contains all the information needed for making proteins (molecules that organisms need to survive). Each protein is encoded by a gene. A gene is the fundamental physical and functional component of

heredity which is a specific sequence of DNA nucleotides that specify how a single protein is to be made. There are two copies of each gene in every individual. One of which is inherited from each parent. The majority of genes in all humans are the same, although there is a minor variation in a small number of genes among humans (< 1 % of the total).

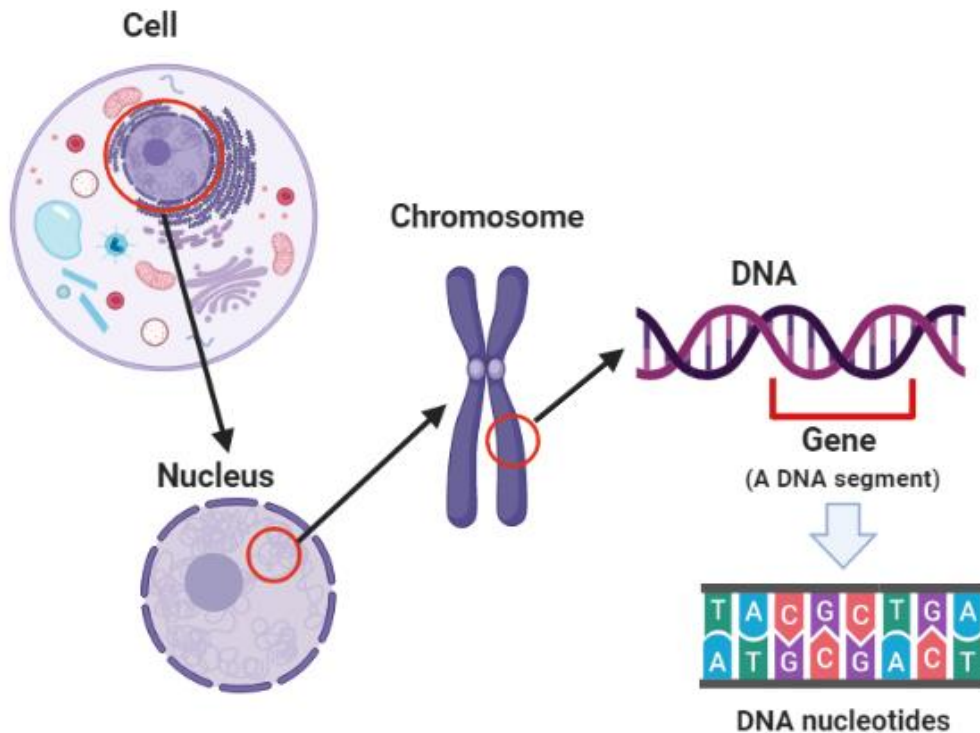


Figure 1. The basic hierarchy within a cell.

## 2.2. Gene Expression

Genes encode proteins and proteins control the function of cells. Thus, the thousands of genes that are expressed in a specific cell decide what this cell will do. An organism cannot use the genes themselves. Gene expression is the procedure that uses genetic instructions to synthesize gene products that carry on important functions such as enzymes, hormones, and receptors.

Gene expression requires two steps: transcription and translation. The transcription process changes the information in DNA to an RNA molecule, which in the case of a protein-coding gene is messenger RNA (mRNA) (Figure 2). The DNA of a gene acts as a basis for complementary base-pairing, and an enzyme known as RNA polymerase II catalyzes the formation of a pre-mRNA molecule, that is subsequently transformed into mature mRNA. The

‘RNA polymerase’ enzyme separates the two DNA strands of a double helix. An mRNA is a single-stranded copy of a gene sequence. Subsequently, the translation process translates the mRNA molecule sequence into a sequence that consists of amino acids during protein synthesis (Figure 2). Furthermore, the cell has a control point for its functions, by changing the quantity and type of proteins it generates, in any stage of the information flow from DNA to RNA to protein. Thus, the expression of many genes can be determined by measuring mRNA (messenger RNA) levels using multiple techniques and gene expression data can give information about the function of previously uncharacterized genes.

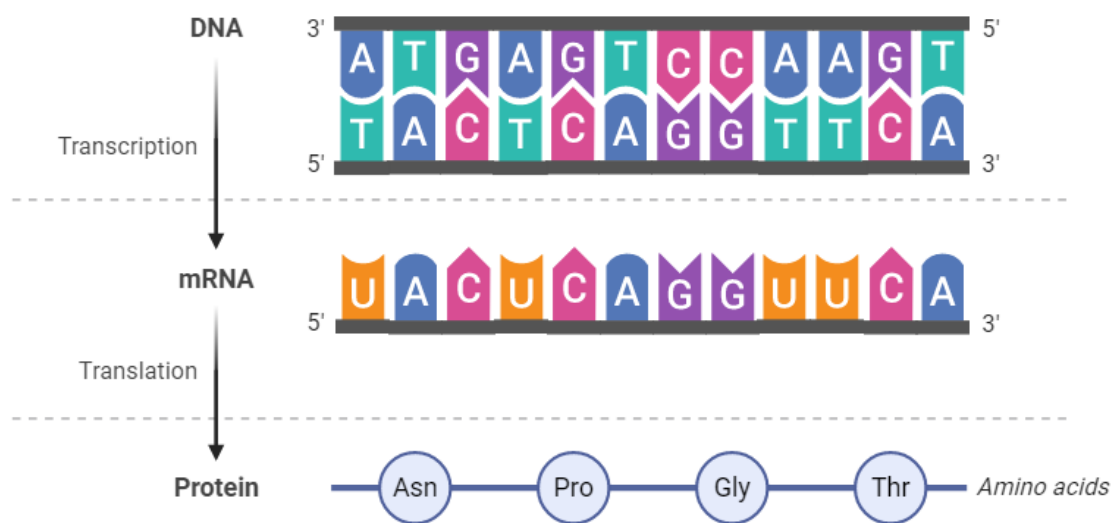


Figure 2. Transcription and translation processes.

### 2.3. DNA Microarray

Numerous genes and their roles have been established in genetic experiments over the last few decades. In addition, it is expected that the human genome project will be completed in the next few years. As genetic data are increasingly available, biological studies have started to move from characterizing only individual components within a biological system to the actions of the biological system in its entirety[28].

Thus, analysis of gene expression has been provided a complementary view of the primary goal in biological and molecular studies in understanding the cell molecular machinery and has been taken an important role in many fields of biological research since changes in the physiology of an organism or a cell will be accompanied by changes in the pattern of gene expression [29].



Science can use many techniques to analyze gene expression by finding out how high or low the expression of a gene is. On the other hand, a significant number of genes cannot be investigated by conventional methods. DNA microarray is one of these proven tools that provide a new approach to the large-scale study of the molecular mechanisms in a cell, or even every gene in an organism, in one single experiment quickly and in an efficient manner. With the extensive technology in DNA chips, they have been used in many areas of biological research such as gene expression profiling, diagnostics, genetic engineering, functional genomics, and DNA sequencing. Typical aims of DNA microarray studies include a diagnostic comparison of the genes that are expressed in different types of cells such as prostate epithelium versus cardiac muscle, as well as in cells subjected to a number of situations, such as, for instance, physical conditions (e. g. temperature, radiation) [30].

DNA microarrays are generated by robotic machines which organize tiny amounts of many gene sequences on a single microscope slide. DNA microarrays often consist of glass slides. The glass slide includes many spots of immobilized DNA (targets) (hundreds to thousands), which could be hybridized at the same time with two samples (probes) of multiple fluorescent coloration colors [28]. The DNA fragments act as probes for specific sequences in a sample, each sequence represents a single gene. In all experiments, RNA is isolated from experimental samples. Note that it is often beneficial to operate with more stable complementary DNA (cDNA) made by reverse transcription at intermediate steps due to the inherent chemical instability in RNA [30]. Experimental RNA samples are converted by reverse transcription (RNA to DNA) into cDNAs labeled with two fluorescent dyes. Before the array is made, however, the cDNA is denatured in order to allow the hybridization of the array. The sample that represents a special condition set up by the experimenter is labeled with a red fluorescent dye (Cy5) and mixed with the reference sample that is labeled with a green fluorescent dye (Cy3). The complementary DNAs (cDNAs) are hybridized with the DNA on the chip (Figure 3). The labeled DNA is only connected to the additional DNA. Microarray is washed and scanned for the two fluorescently labeled cDNAs. In ratio-based analyses, relative intensities of every fluorophore can be used to identify upregulated and downregulated genes [31]. The mRNA value attached to any site in the array indicates the level of expression of the different genes [31]. All data are gathered and a gene expression profile in the cell is established.

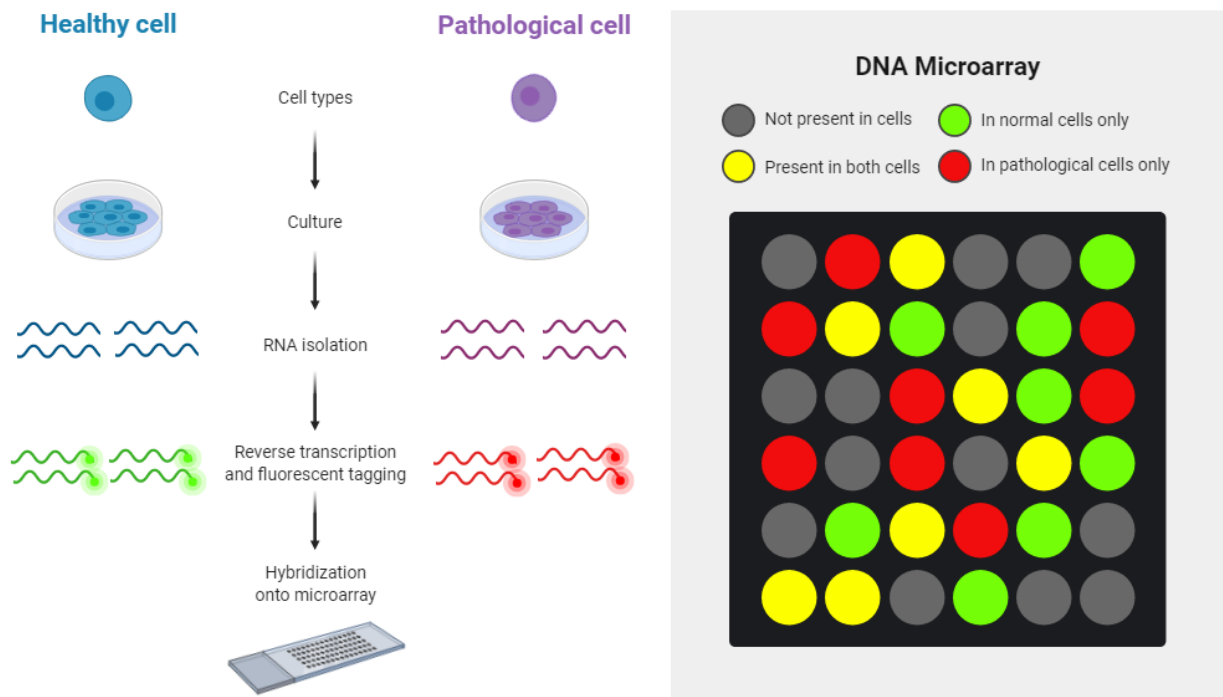


Figure 3. Process flow for gene expression profiles on the DNA microarray.

A sequence of  $n$  level ratios is the product of an experiment on  $n$  DNA samples on one chip [32]. Microarray data from a series of  $n$  separate experiments can be interpreted as a matrix for gene expression with each row consisting of a vector of  $n$  expression values for one gene.

## 2.4. Biological Pathway

For the human body to develop properly and remain healthy, all parts of the body, from individual cells to cells to entire organs, must work together at many different levels. This biological teamwork is made possible by a number of complex and interconnected pathways that promote communication between genes, molecules, and cells. A biological pathway is a sequence of actions between molecules in a cell that results in a certain product or a change in that cell. A pathway can activate the assembly of new molecules, switch genes on and off, or trigger a cell to move. The molecules forming biological pathways interact with each other, as well as with signals, in order to perform their assigned tasks. Biological pathways play significant roles in the development of complex disorders, like cancers, are generally caused by a variety of genetic changes which render pathways not working properly. Analyzing pathways by integrating several types of high-throughput data, like genomics and proteomics, has become one of the key challenges in understanding the mechanisms of complex diseases. Biological pathways exist in several types.

Pathways involved in metabolism, gene regulation, and signal transduction are among the most well-known.

- A metabolic pathway is a chain of linked chemical reactions that occur in human bodies and feed one another. The process by which cells break down glucose molecules in food into energy molecules that can be retained for later use is an example of a metabolic pathway. Other metabolic pathways aid in the formation of molecules.
- Gene regulation involves a wide variety of mechanisms that are used by cells to control which genes, out of the many genes in its genome, are expressed (turned on), or repressed (turned off). Gene regulation also enables cells to rapidly react to changes in their environment. Regulation of genes can happen at any time during gene expression, but most frequently at the transcription level when the gene's DNA is converted to mRNA. Environmental or other cells' signals trigger proteins known as transcription factors. Transcription factors are proteins transcribed by genes and controlled by one or more other transcription factors. Such proteins bind to a gene's regulatory regions and increase or decrease the transcription level. By regulating the transcription level this process can determine the number of protein products produced at any given time by a gene.
- Signal transduction pathway involves the binding of extracellular signaling molecules and ligands that are generated and released by signaling cells, to receptors located on the target cell surface or inside it. The signal moves into the cell after interacting with these receptors, where its message is conveyed by specialized proteins that activate events inside the cell to evoke a specific response. Figure 4 shows an illustration of the WNT signaling pathways.

Over the last 15 years, academic and commercial groups have developed an extensive collection of databases. The information in these databases is extracted from scientific literature or from systematic experiments [33]. Examples are KEGG [34], Reactome [35], WikiPathways [36], NCIPathways [37], Pathway Commons [38], and TRANSPATH [39]. However, these databases vary in terms of their average number of pathways and molecules in each pathway, the biochemical interaction types they involve, and the pathway subcategories [40][41].

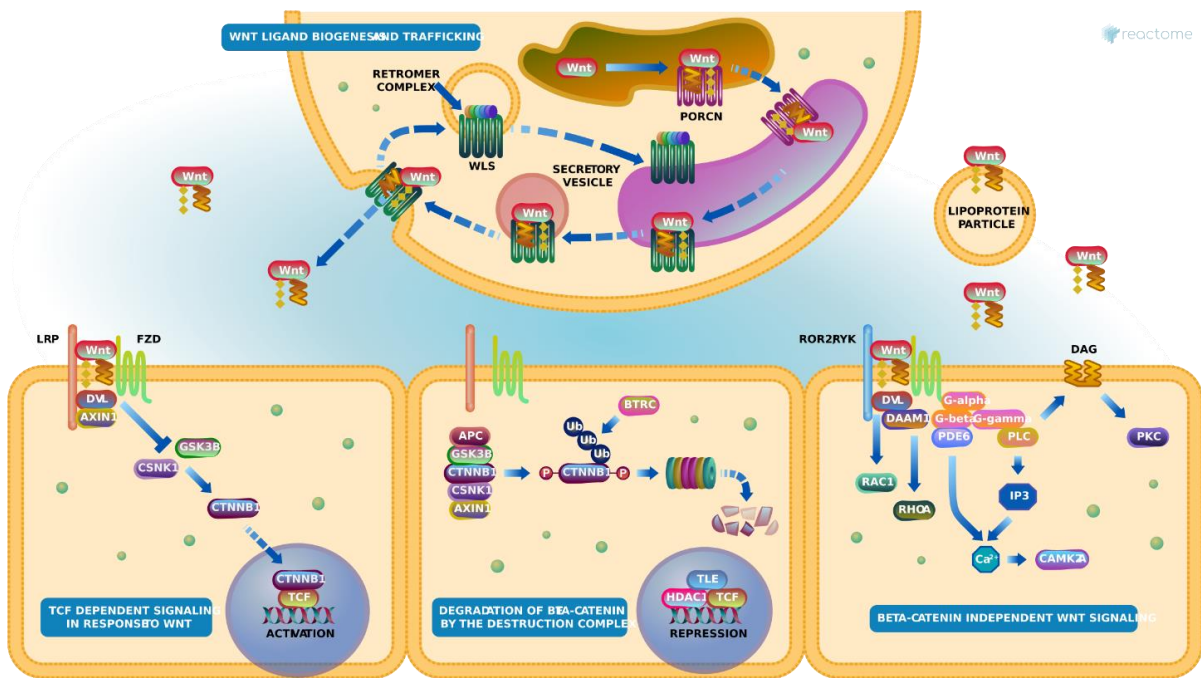


Figure 4. WNT signaling pathways control a wide range of developmental and adult processes in metazoans including cell proliferation, cell fate decisions, cell polarity and stem cell maintenance. (Source <https://reactome.org/content/detail/R-HSA-195721>)

## 3 Bioinformatics Background

### 3.1. Enrichment Analysis

In many genomic, proteomic, or metabolic analyses, the final phase consists of generating a list of biomolecules of interest. Common examples are gene lists ranked by differential or co-expression examined in microarray studies, lists of Single Nucleotide Polymorphism (SNP) genes identified by a genetic link to a particular phenotype in the genome-wide association study and ranked by p-values [42]. These lists usually have no structure and lack meaning. Determining whether the genes interact with others or affect the biological processes being studied is difficult. Vast literature and databases must be examined to answer basic questions like: what is the function of a gene? Does a gene have an interaction with other genes or proteins? Does it behave differently in the process of diseases or treatment? Manual examinations of genes are often unfeasible tasks and time-consuming, particularly on large gene lists. It is not only more biologically intuitive to focus on a set of interesting genes or proteins in its entirety but also has the potential to improve the statistical power [42]. Therefore, it is a crucial task to understand the functional significance of these gene lists.

Enrichment analysis has grown in its potential to give useful insights into the common biological mechanism that underlies a gene list and has become the secondary study of genes from high-throughput genomic techniques. By mapping genes and proteins to their corresponding biological annotations and comparing the distribution of their annotated terms to the background distribution of these terms, enrichment analysis can statistically determine within a list under study the over or under-represented terms which may be associated with disease phenotypes [42]. These enriched terms are assumed to describe some significant biological underlying processes or behavior. Enrichment tools were classified by *Huang et al.* [43] into three classes based on their algorithms: singular enrichment analysis (SEA), gene set enrichment analysis (GSEA) [4], and modular enrichment analysis (MEA). SEA is the most conventional method. It checks on an iterative basis the annotation terms one by one compared to a list of genes that are interesting for enrichment. SEA methods calculate enrichment p-value for each term where the detected frequency of the annotation term is compared with the frequency expected by chance using common statistical methods namely Chi-square [44], Fisher's exact test [45], Binomial probability, or Hypergeometric distribution [46]. Annotation terms are considered enriched when they are beyond a p-value threshold ( $P\text{-value} \leq 0.05$ ). Tools

of this category such as Onto-Express [47], GOSTat [48], and EasyGO [49], rely mainly on the Gene Ontology terms as annotation terms. However, as the SEA independently takes each term, it lacks the hierarchical relationships between relevant GO terms. Such a method often results in lists of hundreds of enriched terms since similar terms are viewed as unique and result in redundancy. Semantic redundancy dilutes the focus on relationships between relevant biological terms among hundreds of other terms. Moreover, a disadvantage of a method that depends on one source of annotation is that it would inherit its limitations. With respect to GO, the annotations are still incomplete and biased against genes that are well-studied.

GSEA-based methods are similar to SEA, however, they include all genes in the study without selecting genes that are considered significant by a threshold. A maximum enrichment score (MES) is computed based on the order of the rank of all members of the gene within a given annotation category [43]. Given a predefined set of genes that share a particular annotation (e.g., genes encoding products in a metabolic pathway), GSEA determines whether those genes are distributed at random over the larger list of ranked genes or primarily appear to be over-represented in the upper or lower section of the longer list of ranked genes [42]. Sets that show the latter distribution, indicate an association with the phenotypic distinction.

MEA uses the same strategy as SEA and considers the term-to-term relationships which may appear during enrichment between annotation terms. The key advantage of this method is that redundancy is reduced, and biological concepts can be prevented from being diluted. Many tools such as Ontologizer [50], topGO [51], and GENECODIS [52], have been recently claimed to enhance sensitivity and specificity by taking into account the relationships between GO terms during enrichment calculations.

In human disease-associated gene or pathway discovery, there are plenty of effective gene set enrichment analysis approaches. Drier et al. (2013) [53], for instance, have shown that enriched gene sets could be used as biomarkers to predict survival time in patients with glioblastoma and colorectal cancer. Zhao et al. [54] combined information of gene set enrichment analysis and microRNA target gene set to identify microRNAs associated with cancer. Lee et al. [55] used gene set enrichment analysis and transcriptional data to identify the driver mutation behind the metastasis of breast cancer. The identification of the enriched gene set would provide valuable knowledge on the molecular functions and underlying mechanisms of various diseases.

## 3.2. Network Biology

Cellular life is a complex network of biological reactions and molecular interactions between active proteins that can be described and explored as the “interactome”. Network representations were used to define interactions in different areas among entities of interest and they are helpful for the analysis and the visualization of complex biological activities [56]. Biological networks are interconnected, as opposed to biological pathways which are series of molecular interactions leading to a final outcome. Network biology is a fast-growing field in biomedical research which reflects the current opinion that complex phenotypes, for example, disease susceptibility, are not triggered by individual gene mutations behaving in isolation but rather the result of the disruption of the gene network. A key to understanding complex systems is to understand the topology of these molecular interaction networks and to recognize molecules that play a key role in structure and regulation. Several different types of relationships can be evaluated in a biological context, like interactions among proteins or genes identified by mutational combinations. Analysis of these networks offers new insights into understanding fundamental mechanisms that regulate normal cellular processes and disease pathologies. The development of high-throughput techniques has allowed components and their biochemical interactions to be established on a large scale. Data obtained from such experiments are often incomplete and contain errors, although useful for the generation of large amounts of biological information. However, valuable information can be given about individual component functions and unexpected interactions between components and cellular processes. The development of high-throughput technologies has established large-scale networks that are accessible from different public databases. Generally, such databases facilitate web-based searches and include rough molecule pair datasets. For protein-protein interactions (PPIs), the most common databases for protein function prediction are BioGRID [57], MIPS [58], and STRING [59]. STRING includes functional interactions of proteins that are identified and predicted with functional similarity scores, thus it provides weighted networks [56].

### **3.3. Biomedical Knowledge Representation and Discovery**

#### **3.3.1. Ontologies for Biomedical Knowledge Representation**

Recently, there has been a significant growth of research in the biomedical informatics field, and a huge amount of research data in the fields of clinical, biomedical, gene research, and patient records has been collected. Simultaneously various biomedical tools have been developed to perform the management of biomedical and clinical research data. The bulk of research data is spread over different databases. These databases are built independently of each other and generated in a wide variety of formats for implementation. Database systems describe objects without giving general concepts and their relations between them. Because of this inconsistency among the research data formats, it has been difficult for clinical researchers to interpret and gather the required data. At this stage, researchers need to represent the knowledge of their domain in a way that defines a common vocabulary of all the relevant concepts and their contexts, in order to share it and reuse it. For this domain, the problem can be solved by semantic technology. An ontology represents semantic knowledge, which provides a common framework for structured concepts, concept definitions, relationships, and axioms in a common language. The design of ontology is a significant task of medical computer science, to interpret the data and acquire inferred knowledge. The use of ontologies started with the development of the Gene Ontology (GO) [3] in around 1998 [60]. Numerous biomedical ontologies have been established in recent years in many domains such as anatomy, medicine, and molecular biology.

##### **▪ Ontology Structure**

An ontology is a formal representation of controlled structured vocabularies that describe concepts (entities) in a certain field of knowledge and their relations. An ontology  $O = (C, R, A)$  consists of defined concepts  $C$  which are interconnected by direct relationships  $R$  (e.g., is-a, part-of...), and described by further attributes  $a \in A$ . Each concept  $c \in C$  is used to reference the concept and has a unique identifier (e.g., id: GO:0000001 from Gene Ontology). A relation  $r \in R$  has a specific type and represents a semantic relation that directly connects two concepts  $c_1$  and  $c_2$ . The concepts of an ontology are typically structured as trees or acyclic graphs, where the concepts represent the nodes and the relations represent the edges (Figure 5).



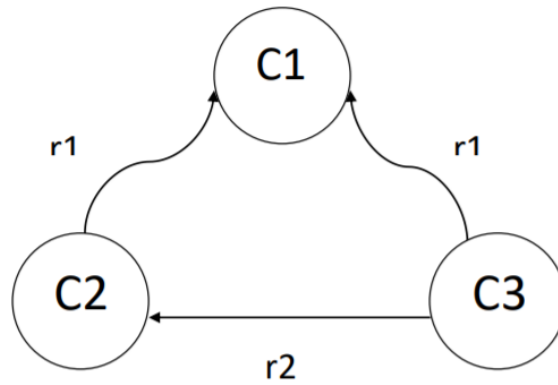


Figure 5. A simple graph represents an ontology with 3 concepts, that are interconnected by two relations (e.g., r1= is-a, r2= part-of).

In an ontology, the concepts of the domain are also defined as classes. The classes can be described in a more specific way as sub-classes. The individuals related to the same class are defined as instances. The attributes that describe the features of a class or an instance are represented and defined as properties (Figure 6).

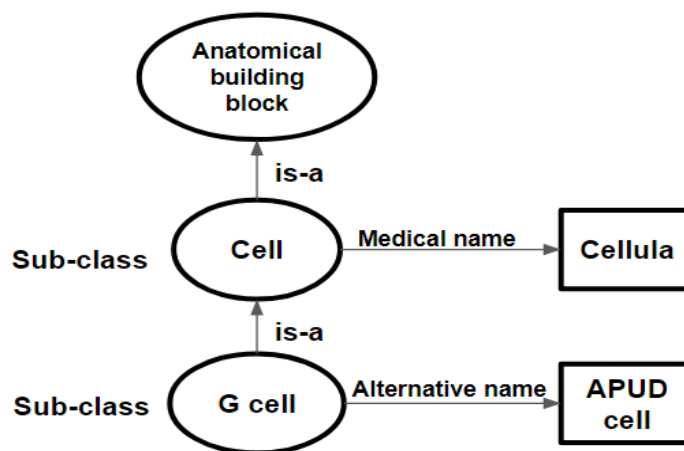


Figure 6. A simple representation of ontology components.

In computer science, an ontology is the working model of entities and relationships in general or in a specific knowledge or practice area, for example, biology or bioinformatics [61]. Ontologies promote data retrieval by enabling annotation grouping and allowing data to be re-

usable according to standards that offer common terminology and structure. The unifying aim of ontologies is “data integration”, either within and across domains, across various species, across granularity levels (organ, organism, cell, molecule), or across different perspectives (medical, biological, clinical).

BioPortal [62] and OBO Foundry [63], are two leading repositories that provide access to a growing number of different biomedical ontologies. These ontologies are classified into different groups: classifications and nomenclatures, phenotype, disease-specific, clinic, anatomy, patient-related data, epidemiology, pharmacy, and health indicators. The ontologies are primarily used in the semantic annotation of different kinds of data objects such as proteins, genes, or literature to achieve a better information exchange. Often different ontologies from one domain containing overlapping or related information are interlinked by ontology mappings (e.g., UML (Unified Medical Language System) [9]). Different powerful ontologies in biomedical and clinical domains are developed by various medical centers, researchers and industries, etc.

Many tools and methods have been developed that allow the use of ontologies and promote their use. These tools often concentrate on one or two of the features of ontologies. Since different, disconnected databases can use the same identifiers, standard identifiers for classes and relations in ontologies have been a key aspect for the integration of data across multiple databases. One of the first applications for which biological ontologies have been created, in particular GO, was making biological sense of the large data sets emerging from the expression array technologies [60]. Particularly, ontologies have allowed the assignment of functions to gene products and their computational comparison within and between organisms.

Moreover, ontologies provide vocabularies that define the concepts and relationships used to represent a field of interest. In the field of health care application, ontologies are used by medical professionals to represent knowledge about diseases, symptoms, and treatments; and by pharmaceutical companies to represent information about drugs. Ontology class labels and relationships allow access to annotated data with these ontologies. For this application type, the integration of this knowledge from the medical and pharmaceutical domain with patient data by an established link provides a way for users to access the information associated with the ontology class. This link then provides the users of an ontology with a way to access the information related to the ontology class and allows a wide variety of applications like decision support tools to look for possible treatments and tools for promoting epidemiological studies.

Ontologies can also help to tackle a challenge facing machine learning and data mining approaches. The use of ontologies can promote the combination of text, structured data, or molecular data in knowledge bases. This can be established by extracting the relevant features from each information type and describing findings using one single ontology that incorporates the information used to train a classifier [60].

### **3.3.2. Text Mining for Biomedical Knowledge Discovery**

Medical and biological studies have evolved to an unprecedented level in recent decades, opening gates to the mechanisms that underlie health and disease in living organisms. It is a major challenge to the research community to incorporate these insights into a unified framework to improve our understanding and decision-making. Effective discovery and development of drugs demand methods to integrate patient data with clinical data, as well as efficient literature mining, to estimate the effectiveness and safety effects of new molecules and treatment strategies. Text mining enables users to update their knowledge on the latest literature, review a wider range of publications, and search for contextual factors that might have become important after the creation of databases. Crucial information on evidence of clinical use of genomic abnormalities is largely reported in biomedical literature. Biocurators, clinicians and oncologists are becoming prohibited from keeping up with the rapidly growing amount of information, particularly the therapeutic implications of biomarkers and therefore relevant for treatment selection. The potential to access the most in-depth information for gene-disease and genotype-phenotype associations is a key factor of precision medicine. Nevertheless, many of the sources required to understand the relationships between genotype and phenotype include unstructured text which is difficult to analyze.

Text mining, also known as text data mining, is the process of analyzing vast amounts of data from unstructured text, in order to convert it into structured data, derive valuable insights and mine knowledge. In a nutshell, it is to mine within the text for something valuable and bring text into a form that is analyzable.

A common definition of text mining is provided by Hearst [64]:

*“Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information to form new facts or new hypotheses to be explored further by more conventional means of experimentation.”*

Text mining uses various computational technologies, including machine learning, natural language processing (NLP) [65], named entity recognition, relationship extraction, and hypothesis generation, to find outcomes hidden in unstructured text. Artificial intelligence (AI) is a computer science field that reveals the need for machine learning (ML) methods in our daily life [66]. Machine learning is the most common approach in the context of text analytics. It is based on a set of statistical and mathematical techniques to identify different aspects of text. Machine learning approaches can automatically analyze data and can recognize hidden patterns from big data, which a human being cannot find [67].

Biomedical research focuses on studying biological processes and investigating the causes of underlying diseases. It is an interdisciplinary domain bridging molecular biology, genetics, and biochemistry with medicine. The study of how various chemical substances control biological processes helps to develop treatments and find new ways to diagnose diseases.

Literature is an essential way to report and publish experimental results, which makes it rich in a large amount of valuable knowledge. With the enormous amount of biomedical literature and the rapid growth of the number of new publications, a huge wealth of scientific knowledge is scattered in multiple biomedical repositories of textual data and research articles. Typically, biomedical knowledge is largely represented in text using natural language. Extracting relevant information and analyzing text data is helpful to discover relationships between biological entities such as gene-to-disease and disease-to-drug associations and to generate new hypotheses.

The quest for literature is also a core component of finding relevant information in any new scientific discovery process. Due to its exponentially growing scale and interdisciplinary existence, biomedical literature is unique. It is not only a tool to find the newest answers but is also a record of what researchers have concerned themselves with over the decades [68]. Access to biomedical literature is crucial for multiple user types particularly clinicians, bioinformaticians, and biomedical researchers. The biomedical field has been greatly supported by the efforts of the US National Library of Medicine in order to provide bibliographic material for most journal papers, particularly abstracts [69]. PubMed [70] has been the most commonly available research method devoted to life sciences and biomedical literature [71]. PubMed does a thorough job of covering this vast literature, as it contains citations from all around the world. PubMed is managed by the National Library of Medicine (NLM), which belongs to the US National Institutes of Health (NIH) [72].

The volume of biomedical literature in electronic format is growing with the ease of Internet access [73]. PubMed/MEDLINE® includes millions of citations from MEDLINE®, life

science journals and online books for biomedical literature (Figure 7) [74]. MEDLINE® [75] is the National Library of Medicine (NLM) journal citation database. PubMed has been the key resource for searching and retrieving biomedical literature electronically since its foundation [76][77].

A large number of biomedical text-mining studies have focused solely on processing abstracts from PubMed. It is advantageous to work with abstracts, since they are publicly and freely available, and summarize the main points of their associated articles, which makes them rich in information content. MeSH® (Medical Subject Headings) [78] is the controlled vocabulary system used to index PubMed articles. The MeSH® terms provide constancy for biomedical literature indexing which facilitates the retrieval of relevant articles. It contains distinct types of terms that help to improve the search results. Additional state-of-the-art biomedical search tools are available that provide indexing biomedical articles such as Embase (Excerpta Medica dataBASE) [73], a database of biomedical citations for multipurpose use, and others that are not commonly used.

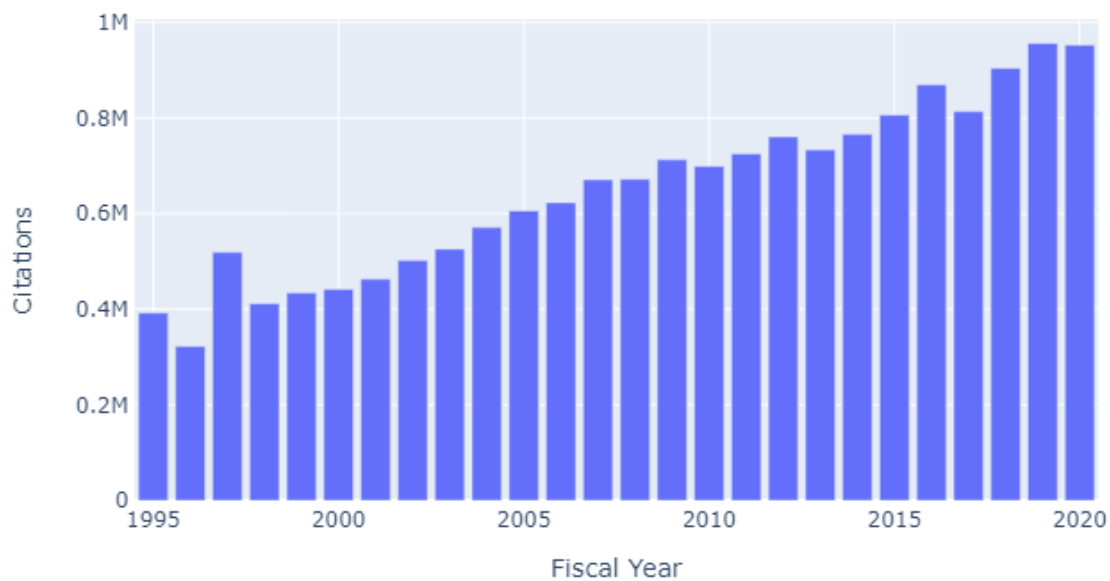


Figure 7. The number of indexed<sup>1</sup> citations that have been added to MEDLINE since 1995 in a fiscal year [77].

<sup>1</sup> “Indexed citations are those citations selected for MEDLINE that have completed processing and indexing with current MeSH® (Medical Subject Headings®). Indexed citations have a status of MEDLINE” [77].

## 4 Enrichment Analysis Tool

### 4.1. Introduction

A common statistical method to interpret the function of the genes that are differentially expressed is to assign biological meaning to them and determine which predefined sets of genes are statistically significant among a list of ranked genes. Enrichment analysis methods were introduced in section 3.1.

LRpath is a logistic regression-based method that was introduced and implemented by *Sartor et al.* [79] with the motivation of finding an optimal approach to identify predefined gene sets that are biologically related and enriched with genes that are differentially expressed. The LRpath function works by detecting the variation of the probability of a gene that belongs to a predefined set of genes (dependent variable) compared to the significance of differentially expressed genes (independent variable) [79]. LRpath's primary question is whether “the odds of a gene belonging to a predefined gene set increase as the significance of differential expression increases” [79]. The approach models the probability of a randomly selected gene that belongs to a particular category given the level of significance of that gene. For categories significantly affected by the experimental condition, the probability increases as the statistical significance increases. Comparisons were made with other similar approaches using experimental and simulated breast cancer datasets. Based on all criteria, LRpath has performed better than the other tested approaches. More information about the functionality of LRpath can be found in the original paper [79]. LRpath is implemented as an R function. The method is designed to assess GO terms and KEGG pathway categories using ENTREZ gene identifiers, but it can be modified for other categories. Using the novelty of this function, we implemented a Java tool that integrates the function in R with a Java interface in order to integrate it into the geneXplain platform [80], which is a Java-based software platform, and to make it accessible for Java users. Moreover, we used our own data sets that include GO terms and Reactome pathways, mapped to Ensembl [81] gene identifiers to test categories. We used the Biomart [82] interface to extract the mappings of Ensembl IDS and GO/Reactome identifiers. The GO terms in the mappings are only the terms that are at the lowest level of the ontology, and enrichment analysis tools generally use all the terms in the Gene Ontology to test categories. Therefore, we used the Ontology-Based Answers (OBA) [27] service to access the ontology

structure and map the Ensembl IDs to all the terms in the ontology. We will describe in more detail in the following sections the processing methods we applied.

## 4.2. Materials and Methods

### 4.2.1. Logistic Regression Approach

Logistic Regression is a statistical method for predicting a binary outcome using the log of odds as a dependent variable and one or more independent variables. Logistic regression fits data to a sigmoid function that takes real-valued numbers and returns a probability value between 0 and 1. According to a given cutoff, the output can be classified as positive or true if it is above the cutoff value, or as negative or false if it is less than the cutoff value.

### 4.2.2. Gene Ontology

Gene ontology (GO) [3] is an ontology that represents the knowledge of the biological domain. It is a structured, controlled terminology of ‘terms’ describing gene product properties across all organisms. Gene Ontology is the primary source of information on the function of a gene, the biological processes in which it plays a role and the location of the cell in which it is located. It consists of sub-ontologies that cover three domains which are: cellular component, molecular function, and biological process [3].

- Cellular component: describes the parts of a cell where gene products are located.
- Molecular function: describes a gene product's essential activities at the molecular level.
- Biological process: describes pathways and sets of molecular events made up of the activities of multiple gene products.

The concepts in the ontology are related to each other with three types of relationships: “is\_a” (a subtype of), “part\_of”, and “regulates”.

- **“is\_a”**: A “is\_a” B means term A is a subtype of term B.
- **“part\_of”**: A “part\_of” B means B (parent) is a broader term, and A (child) is a more specific term. And all instances of A exist as part of instances of B.
- **“has\_part”**: means that the subject of the assertion must necessarily have the parts referred to by the relationships.

The three listed sub-ontologies are “is\_a disjoint” which means there is no “is\_a” relationship between every node in the three ontologies. Nevertheless, other relations like “regulates,” may link nodes from different sub-ontologies [2].



The GO provides two main resources: the GO itself [83] and the GO annotations [84]. The GO ontology structure can be represented as a hierarchical directed acyclic graph (DAG) of GO terms and the relationships between them. Each term of the GO graph is a node, and the relationships between the nodes are edges. A descendant (child) term is more specialized than its ancestors (parents) and can have more than one ancestor (Figure 8). Whereas the descendant may have only one ancestor in the hierarchal layout.

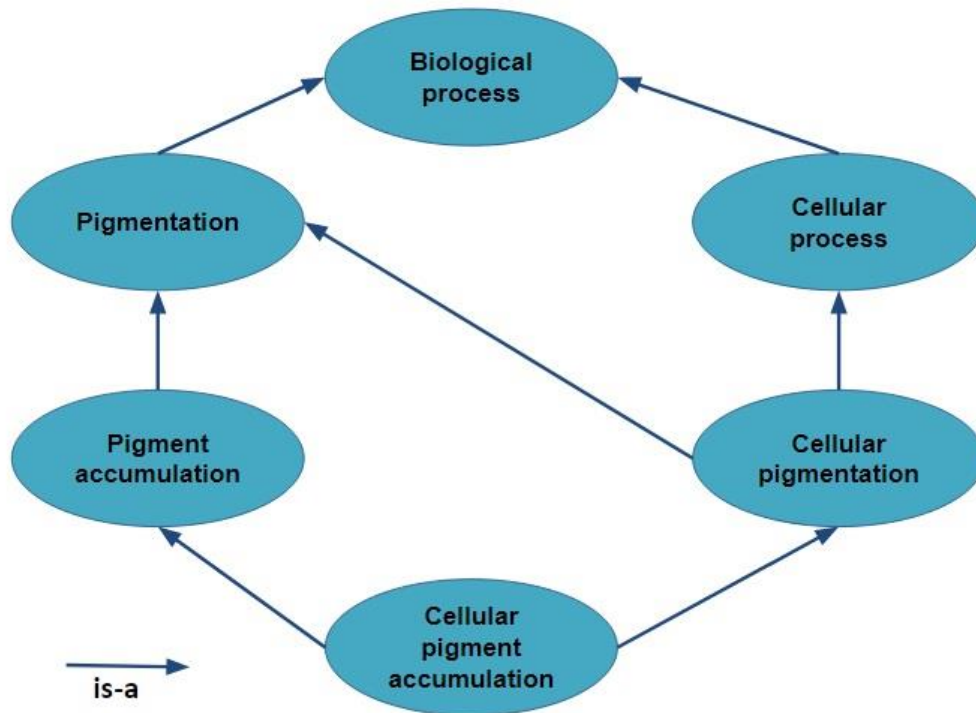


Figure 8. A simple representation of a couple of GO terms related by the "is\_a" relationship.

A GO annotation is a description of a particular gene function. A GO annotation is generated from an association between a term from the ontology and a gene or a gene product; and the evidence which supports the association. Expert curators assign GO annotations either from experimental annotations or from curated annotations. Experimental annotations are based on primary literature experimental evidence. Curated annotations are based on evidence of sequence similarity, review papers, and database entries. GO has become the most common annotation source and the main widely source to perform enrichment analysis on gene sets and functionally interpret experimental data. Moreover, GO provides mappings that consist of GO

terms that are cross-referenced to related concepts from a range of external vocabularies such as KEGG, Reactome pathways, and Wikipedia.

### **4.2.3. Ontology-Based Answers Service (OBA)**

The Ontology-Based Answers service (OBA) [27] is an application that offers the user access to hosted ontologies and functions to answer ontological queries based on the structure of the ontologies. There are many tools that process ontologies, e.g. OntoCat [85], Bioportal [86], and Protégé [87]. Nevertheless, they share two common aspects: addressing access to ontologies in a generic way, and ignoring the unique roles of individual ontologies, dismissing the encoded information ignored in the various relationships [27].

The OBA service uses a client-server architecture. The OBA server is responsible for loading ontologies in OBO and OWL formats. It uses special ontologies' functions implemented in plugins, called 'Semantic Functions'. The server uses already existing plugins such as the Tribolium plugin for the Tribolium anatomical ontology (TrOn) [88]. The service can be extended with new plugins by implementing new semantic functions related to specific ontologies.

Besides the provided Java client, the client can be a web browser, a command-line client or any custom client. Using one of these clients, the OBA service can be embedded into any custom application. Existing projects already use the service such as EndoNet [89] and iBeetle [90] projects.

The server communicates with the client using the Representational State Transfer (REST) interface and produces the answer in three different formats: "text/plain", "text/HTML" and "application/JSON" (MIME types).

The OBA service is available at <http://www.bioinf.med.uni-goettingen.de/resources/oba/>.

### **4.2.4. Reactome**

Reactome [91] is a curated human pathway data resource that offers a computational infrastructure across the biological reaction network. Reactome knowledgebase provides an organized molecular transformation network that includes descriptions of signal transduction, DNA replication, metabolism, and other cellular processes in one consistent data model [92]. It also provides mapping lists that consist of tables of external protein, gene or small molecule

identifiers in a source database, e.g. UniProt [93], Ensembl [81], NCBI Gene [94], or ChEBI [95], which are mapped to Reactome pathway and reaction annotations.

#### 4.2.5. Ensembl

Ensembl [81] is a well-known database source to retrieve annotated genome information. It provides a rational way of organizing genes, transcripts, and proteins. Ensembl uses stable IDs to label the features, such as genes, transcripts, exons, or proteins. The stable identifiers (IDs) are generated from species annotation for the first time. Unlike gene names, which can change as a result of improvements in scientific knowledge, stable identifiers should continue to refer to the same genomic features [96]. A stable ID is composed of four parts:

- The “ENS” prefix is a standard for all IDs.
- A “species prefix”: refers to what species they are in, **e.g.** “MUS” for mouse species. For the human species, it is only “ENS”.
- “feature type prefix”: refers to feature type **e.g.** for gene it is “G” and for transcript it is “T”.
- A unique eleven-digit number: that identifies the entity.

A full Ensembl ID example: “ENSG00000162367”.

#### 4.2.6. Biomart

BioMart [82] is a generic interface that enables scientists to perform advanced queries via a broad number of biological databases. BioMart is a generic and scalable system that is easy to operate and is thus integrated with major data resources such as Ensembl, HapMap, UniProt, MSD, PRIDE, Dictybase, and Reactome [82].

BiomartR [97] is an R package that enables direct access to retrieve a large amount of data without referring to the complex schemas of the databases. The BiomartR package provides easy functions for the collection of all or selected genomic, proteomic data [97]. It also provides the access functions to retrieve the mappings of databases between each other. BioMart provides functional access to the mappings of Ensembl gene IDs to GO and Reactome identifiers using different datasets e.g. Humans, Mouse, and Rat. The terms are actually mapped to Ensembl IDs on the transcript level and through the UniProt IDs.

### 4.2.7. Data Sets Processing

Enrichment analysis methods are mainly dependent on gene/protein annotations provided by knowledge bases such as Gene Ontology (GO) to identify which annotations are enriched and to assign functions to a list of genes/proteins obtained from high-throughput analysis. In the LRpath study, they used the ENTREZ [98] gene identifiers that are annotated to Gene Ontology and KEGG [99] identifiers. In our tool, we used the Ensembl gene identifiers. For the annotation sources, we used Gene Ontology and Reactome. Using the “attributes” feature from Biomart in R ("ensembl\_gene\_id", "go\_id" and 'reactome'), we extracted the mappings of GO/Reactome IDs and Ensembl gene IDs. For each database mapping, we selected three species datasets: Human (Hs), Rat (Rn), and Mouse (Mm).

One Ensembl gene identifier is mapped to multiple GO or Reactome identifiers since a gene can play a role in multiple biological processes/pathways and can be located in different cellular components (Figure 9).

ensembl_gene_id	go_id	ensembl_gene_id	reactome_id
ENSG00000198888	GO:0005747	ENSG00000198888	R-HSA-1428517
ENSG00000198888	GO:0044281	ENSG00000198888	R-HSA-611105
ENSG00000198888	GO:0005515	ENSG00000198888	R-HSA-163200
ENSG00000198888	GO:0022904	ENSG00000198888	R-HSA-1430728
ENSG00000198888	GO:0008137	ENSG00000198763	R-HSA-1428517
ENSG00000198888	GO:0006120	ENSG00000198763	R-HSA-611105
ENSG00000198888	GO:0005743	ENSG00000198763	R-HSA-163200
ENSG00000198888	GO:0031966	ENSG00000198763	R-HSA-1430728
ENSG00000198888	GO:0044237	ENSG00000198804	R-HSA-3700989
ENSG00000198888	GO:0055114	ENSG00000198804	R-HSA-5628897
ENSG00000198888	GO:0016020	ENSG00000198804	R-HSA-1428517

Figure 9. The mappings of Ensembl gene identifiers and GO/Reactome identifiers for Human species. The “R-HSA” prefix in Reactome identifiers represents identifiers that are specific to Human species.

In order to prevent the delivery of data through Biomart every time a user gets access to the application; we stored the mapping tables in a MySQL [100] database. We connected the database to the R functions using the ‘RMySQL’ (Database Interface and ‘MySQL’ Driver for

R) [101] library in R to fetch the mapping data. The mappings were saved in tables, where each table corresponds to one species and one annotation source (Figure 10).

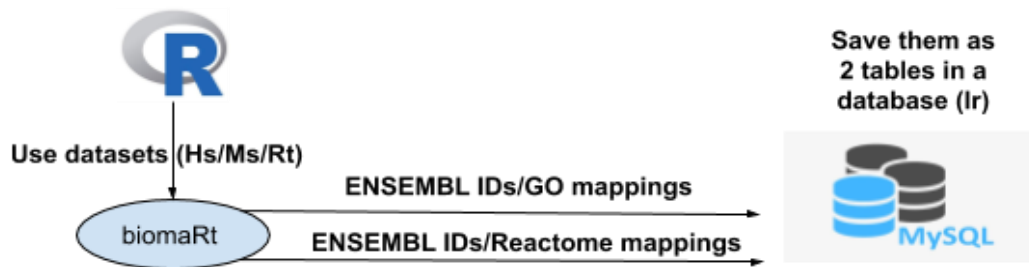


Figure 10. The workflow of extracting and storing the mappings.

In order to enable the integration of the enrichment function into the geneXplain platform, a Java-based platform, and other Java-based applications, we implemented a Java wrapper interface. The Java interface allows the user to interact with the tool as a Java application. We connected arguments between R and Java using the “RCaller” [102] (a software library for calling R from Java) library in Java. We will show the tool functionality in the ‘Results’ section 4.3.

#### 4.2.8. GO Plugin in OBA

Another main feature for an enrichment tool that uses an ontology as an annotation source, is to use all the concepts which are in the ontology structure. However, the mappings of the Ensembl and Gene ontology identifiers extracted from Biomart, cover genes that are annotated to the most specific terms or which are at the lowest level of the ontology graph. To address this issue, we needed to extend the mappings to the terms that are at higher levels so that we cover all the terms in the ontology.

The OBA service can load plugins that allow applying specific functions to specific ontologies. By using this feature, we implemented a new GO plugin that provides access to Gene Ontology and retrieves ontologies’ specific information. Figure 11 depicts the process of how we used the GO plugin in OBA to generate mappings at all the ontology levels.

# Pre-Processing

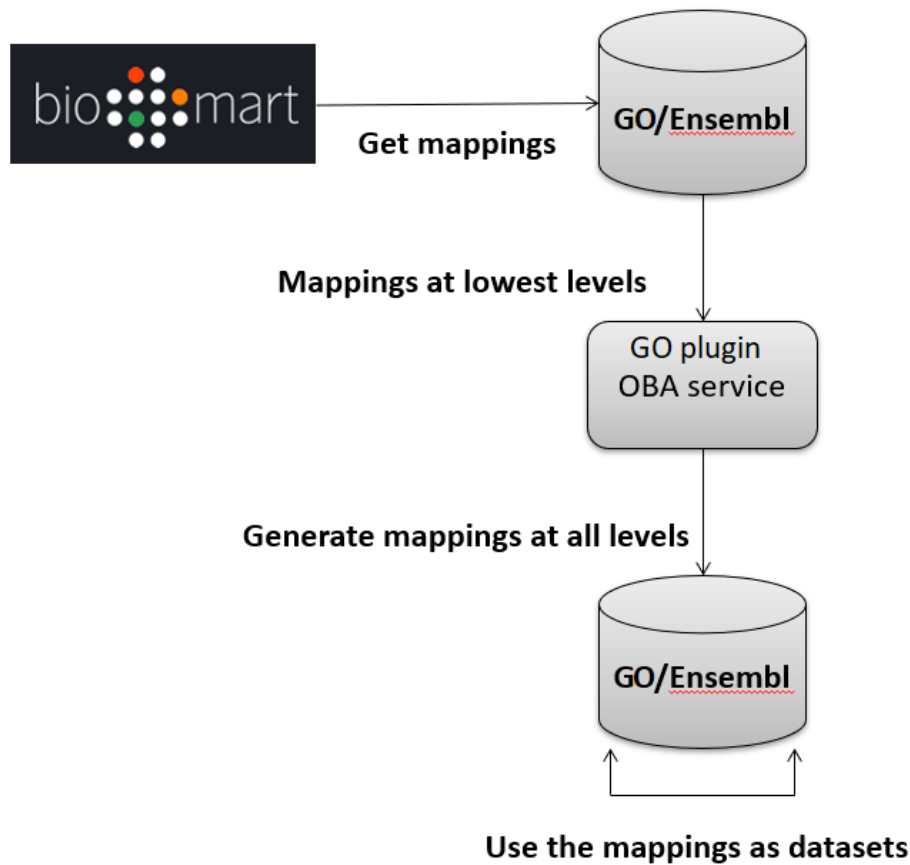


Figure 11. The process of generating mappings of Ensembl/Gene ontology identifiers extracted from Biomart at all the GO levels using the GO plugin in OBA.

The plugin was used to calculate for each GO term all the ancestors, on all paths to root. The ancestors are calculated using special semantic methods built in the OBA server that can walk over the classes of an ontology in downstream or upstream search. After calculating these ancestors for each GO term in the database, the ancestors should also be mapped with Ensembl Gene IDs, so they can be used later as predefined datasets in the enrichment tool. Each term of the ancestor is mapped to the same Ensembl gene ID mapped to its descendant (the GO term to which we have calculated the ancestors) (Figure 12), and the mapping result will be added to the corresponding database species table.

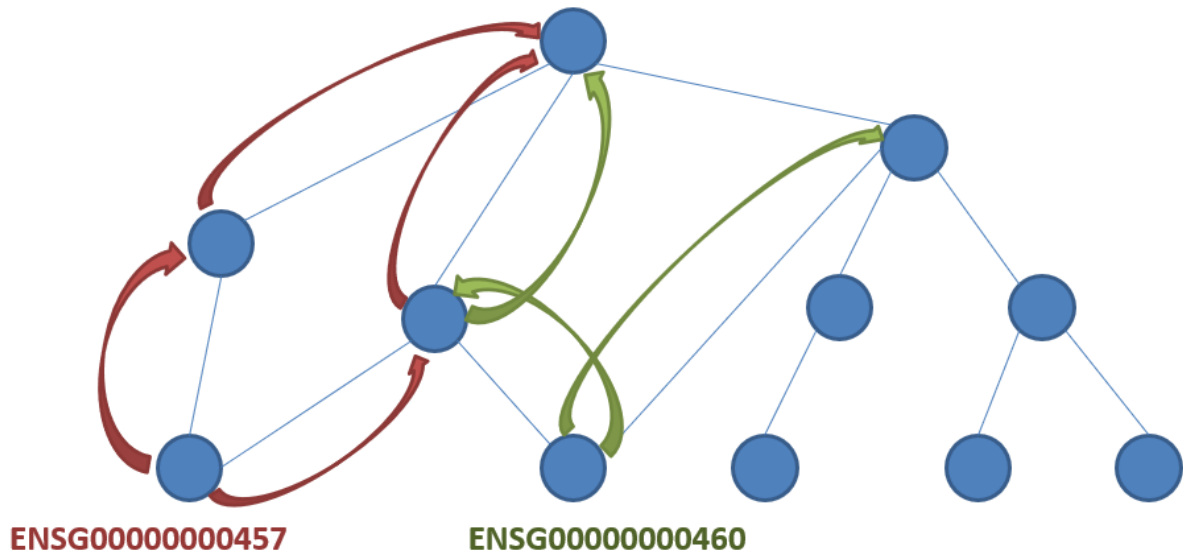


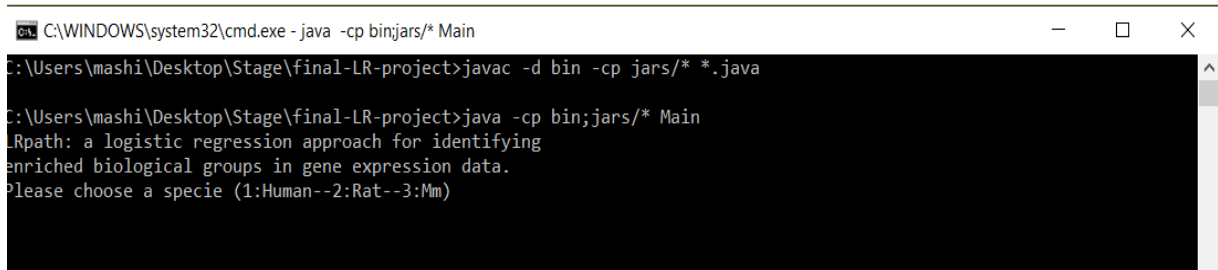
Figure 12. Each Ensembl gene ID mapped to a GO term is mapped to all its ancestors.

After applying this process for all available GO terms in our database, and for the three species (Human, mouse, and rat), we ended up with a database that contains mappings between the Ensembl Gene IDs and all GO terms of the Gene Ontology. The stored mappings were used in our enrichment tool as predefined datasets.

### 4.3. Results

- **LRpath Java Tool Architecture**

An application user interface should be characterized by being simple, straightforward, and provides easy access to common features or commands. We implemented a Java interface that uses the logistic regression function in R in the background by bridging Java and R functions. The interface allows the user to interact with the tool as a Java application that can be used in two different ways: a command-line interface and a Java-point project. With a command-line interface, the user can communicate with the application by typing commands. The user starts the application by running the jar file of the project. When the jar file is compiled successfully, the user will be asked to enter the input parameters progressively (Figure 13).



```
C:\WINDOWS\system32\cmd.exe - java -cp bin;jars/* Main
C:\Users\mashi\Desktop\Stage\final-LR-project>javac -d bin -cp jars/* *.java
C:\Users\mashi\Desktop\Stage\final-LR-project>java -cp bin;jars/* Main
Lrpath: a logistic regression approach for identifying
enriched biological groups in gene expression data.
Please choose a specie (1:Human--2:Rat--3:Mm)
```

Figure 13. Command-line interface.

A Java-point project can be imported with the source code into any Java IDE (integrated development environment) platform such as Eclipse [103], NetBeans [104], and IntelliJ IDEA [105]. The project should be set up by adding the required libraries. Once the project is configured, the user can compile it and the project will be ready to be run or debug in the IDE. The application features provide the ability to integrate it into any Java-based platform.

The application uses the terms of a database and their annotations as predefined gene sets. It is available for GO and Reactome databases and for three different species (human, rat, and mouse), but it can be extended to include more or different databases and species.

To perform the analysis, the user should define the following variables:

- The species of the data set (human, rat, or mouse)
- The database (GO or Reactome),
- The input file contains the list of gene IDs (the gene IDs should be Ensembl gene IDs) with continuous significance values (i.e. p-values).

The input file should be an excel file that consists of two columns, one for the gene IDs and one for the p-values, as shown in Figure 14.



	A	B
1	ID	P.Value
2	ENSG00000067082	6.75E-05
3	ENSG00000073756	5.19E-05
4	ENSG00000081041	1.41E-08
5	ENSG00000089692	9.01E-05
6	ENSG00000099860	7.91E-05
7	ENSG00000100906	7.32E-08
8	ENSG00000107968	1.09E-05
9	ENSG00000108551	8.63E-05
10	ENSG00000112149	2.39E-06
11	ENSG00000115009	3.96E-08
12	ENSG00000118503	7.74E-06
13	ENSG00000118523	9.06E-05
14	ENSG00000120129	6.99E-05
15	ENSG00000122877	1.64E-07
16	ENSG00000123358	3.18E-05
17	ENSG00000125740	3.71E-09
18	ENSG00000128016	6.59E-05

Figure 14. Input file sample.

The tool then retrieves the mappings of the given Ensembl gene IDs to the terms belonging to the given species in the corresponding database. Only the gene IDs with mappings existing in the database are used. After that, the tool connects to R to apply the logistic regression function. LRpath checks for gene sets (termed concepts) that have significantly higher values of significance (for differential expression) than expected at random [106]. Once LRpath has completed an analysis, a new p-value for the significant terms and the significant gene IDs belonging to each term is produced. These p-values are used to identify the most enriched genes, for example, the terms having a value less than the commonly used thresholds ‘0.05’ or ‘0,01’, are considered as the most significant terms. The architecture of the whole application is shown in Figure 15.

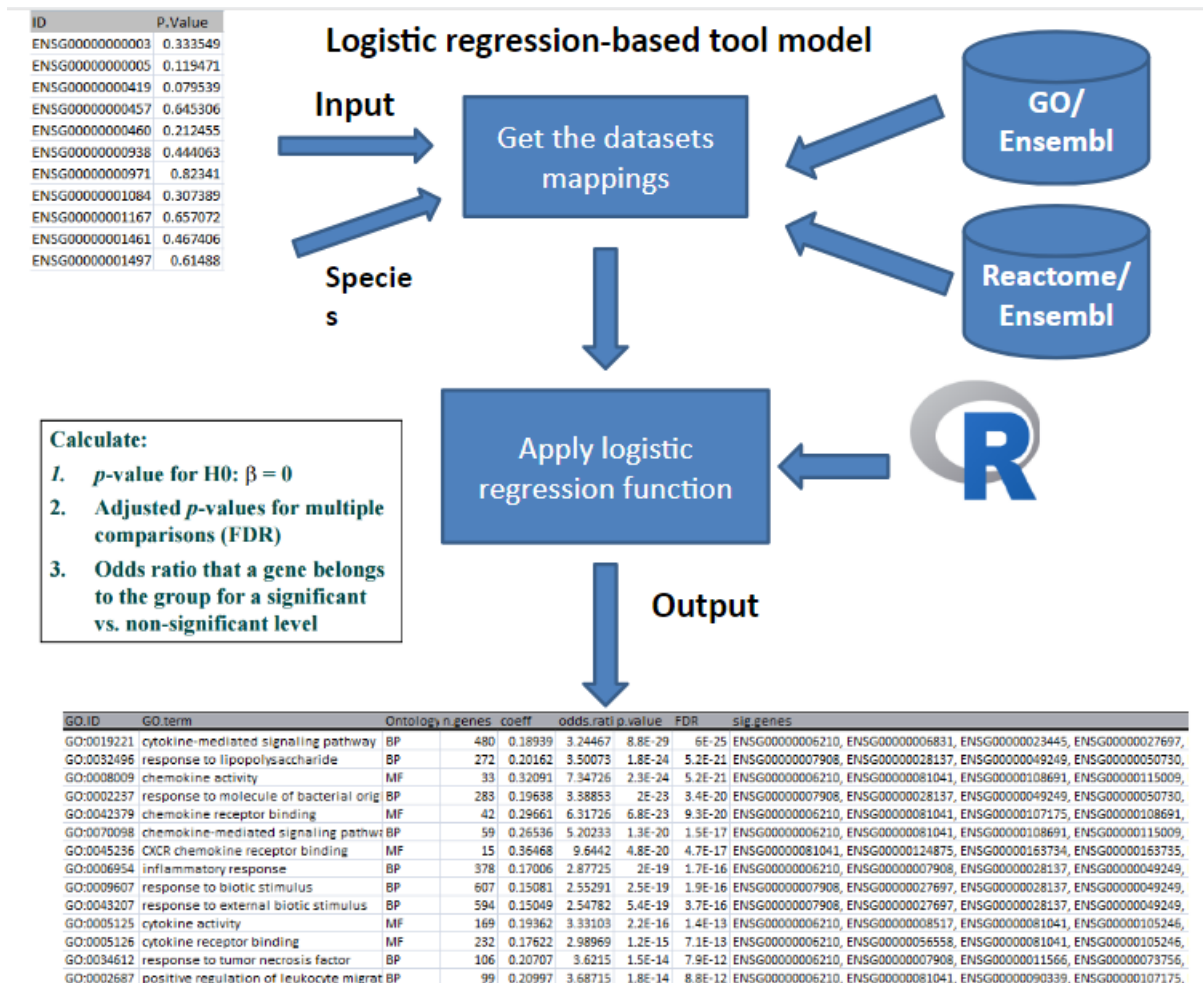


Figure 15. LRpath Java Tool Architecture.

We applied the tool to a data sample to test its functionality. The data were tested with Gene Ontology terms. The analysis was performed successfully, and it generated the results shown in Figure 16. The results are provided in a tabular format. They include the concept names, the concept type, the number of genes making up the concepts, the coefficient, odds ratio, p-value, FDR (false discovery rate), and a list of the significant genes.

GO.ID	GO.term	Ontology	n.genes	coeff	odds.ratio	p.value	FDR	sig.genes		
GO:00192	cytokine-mediated signaling pathway	BP	480	0.189395	3.244672	8.79E-29	6.02E-25	ENSG00000006210, ENSG000000		
GO:00324	response to lipopolysaccharide	BP	272	0.201617	3.500733	1.83E-24	5.2E-21	ENSG00000007908, ENSG000000		
GO:00080	chemokine activity	MF	33	0.32091	7.34726	2.28E-24	5.2E-21	ENSG00000006210, ENSG000000		
GO:00022	response to molecule of bacterial origin	BP	283	0.196375	3.388529	2.01E-23	3.44E-20	ENSG00000007908, ENSG000000		
GO:00423	chemokine receptor binding	MF	42	0.296605	6.31726	6.83E-23	9.35E-20	ENSG00000006210, ENSG000000		
GO:00700	chemokine-mediated signaling pathway	BP	59	0.26536	5.202328	1.34E-20	1.53E-17	ENSG00000006210, ENSG000000		
GO:00452	CXCR chemokine receptor binding	MF	15	0.364682	9.644204	4.77E-20	4.66E-17	ENSG00000081041, ENSG000000		
GO:00069	inflammatory response	BP	378	0.170057	2.87725	2E-19	1.71E-16	ENSG00000006210, ENSG000000		
GO:00096	response to biotic stimulus	BP	607	0.150812	2.552915	2.47E-19	1.87E-16	ENSG00000007908, ENSG000000		
GO:00432	response to external biotic stimulus	BP	594	0.15049	2.547815	5.37E-19	3.67E-16	ENSG00000007908, ENSG000000		
GO:00051	cytokine activity	MF	169	0.193621	3.33103	2.18E-16	1.36E-13	ENSG00000006210, ENSG000000		
GO:00051	cytokine receptor binding	MF	232	0.176225	2.989689	1.24E-15	7.06E-13	ENSG00000006210, ENSG000000		
GO:00346	response to tumor necrosis factor	BP	106	0.207075	3.621496	1.5E-14	7.92E-12	ENSG00000006210, ENSG000000		
GO:00026	positive regulation of leukocyte migration	BP	99	0.209966	3.687152	1.8E-14	8.82E-12	ENSG00000006210, ENSG000000		
GO:00340	response to cytokine	BP	319	0.156539	2.645415	3.49E-14	1.59E-11	ENSG00000006210, ENSG000000		
GO:00431	regulation of I-kappaB kinase/NF-kappaB	BP	195	0.175801	2.98183	7.62E-14	3.26E-11	ENSG00000023445, ENSG000000		
GO:00026	positive regulation of leukocyte chemotaxis	BP	73	0.220663	3.940594	1.18E-13	4.77E-11	ENSG00000006210, ENSG000000		
GO:00705	response to interleukin-1	BP	87	0.208338	3.650033	5.94E-13	2.26E-10	ENSG00000006210, ENSG000000		
GO:00026	regulation of leukocyte chemotaxis	BP	87	0.206672	3.61244	1.2E-12	4.33E-10	ENSG00000006210, ENSG000000		
GO:00313	regulation of defense response	BP	648	0.122691	2.143585	1.52E-12	5.19E-10	ENSG00000003989, ENSG000000		
GO:00716	regulation of granulocyte chemotaxis	BP	37	0.248708	4.690886	1.92E-12	6.17E-10	ENSG00000006210, ENSG000000		
GO:00712	cellular response to lipopolysaccharide	BP	110	0.19445	3.348222	1.98E-12	6.17E-10	ENSG00000028137, ENSG000000		
GO:00026	positive regulation of immune system process	BP	758	0.116076	2.057241	3.12E-12	9.27E-10	ENSG00000006210, ENSG000000		
GO:00313	positive regulation of defense response	BP	347	0.14281	2.429068	6.19E-12	1.77E-09	ENSG00000006210, ENSG000000		
GO:00069	defense response	BP	1203	0.100593	1.86852	6.86E-12	1.88E-09	ENSG00000005249, ENSG000000		
GO:00022	pattern recognition receptor signaling pathway	BP	129	0.183737	3.132563	8.46E-12	2.23E-09	ENSG00000023445, ENSG000000		

Figure 16. A results sample of the LRpath Java tool.

## 4.4. Discussion

The detection of predefined sets of genes that are enriched with DEGs has become a standard aspect of microarray analysis and gives more biological insights to investigators than just significant gene lists. The method developed by *Sartor et al.*, LRpath, uses the logistic regression model to relate the membership and the differential expression of gene sets in respect of enrichment probability ratios [79]. The fundamental question, LRpath answers, is that the odds of a gene that belongs to a predefined group of genes increase with the increase in the significance of differential expression. In comparison to  $\chi^2$ -type approaches, the method enables data arising from experiments of differential expression to stay on a constant scale. In this way, the requirement for the selection of significance cutoff is eliminated and has the benefit of considering the significance level distribution for genes that do not belong to or belong to the gene set under study. If the experiment affects gene expression that occurs in a particular biological pathway, it would be expected that genes with significant P-values are most probably to be part of this pathway than genes with less significant P-values, even it might not be known precisely where a line has to be drawn between the differential expressions: non-significant and significant [79]. In-depth comparisons were made with other related approaches using simulated and experimental breast cancer datasets. On the basis of all criteria, LRpath has overall better performance than the other tested methods. LRpath was implemented as an R function. The function is implemented to check the GO terms and KEGG pathway categories. However, it can be changed to be used with other categories specified by users. We used the function to develop a Java-based tool. The tool is designed to be used using a Java interface. We integrated the LRpath function in R using a Java library that facilitates the exchange of parameters from R to Java and the other way around. The original function uses the ENTREZ gene identifiers for the gene list to be examined and for the predefined gene sets. In our tool, we used the Ensembl gene identifiers. For the categories, we used GO terms and Reactome pathways. We obtained the mappings of Ensembl gene IDs with GO/Reactome IDs for 3 different species: human, mouse, and rat, from the Biomart interface in R. We stored the mappings in a MySQL database and implemented functions in R to call the database and fetch data. The analysis result is a list of the enriched categories with more information about these categories (p-value, genes in these categories...) saved in an excel file. The tool is a standalone Java application that is implemented as a command-line interface and as a Java project that can be imported into Java platforms. This facilitates the integration of the tool into any other Java

platforms. The user still has the option to use the categories he wants by getting the mappings of the gene IDs with any other category source and save them in the database.

Enrichment analysis tools are diverse in their analysis methods and their accessibility (web tools, R functions...). Providing tools with different categories, different gene identifiers, and diverse ways of use, can also help researchers to give them more analysis options and to facilitate the enrichment analysis in the way they prefer.

## 5 Biomedical Knowledge Integration

### 5.1. Introduction

In general, biologists and physicians generally need to compare their biological and clinical data to existing knowledge in databases in order to interpret experimental data. For example, users might want to get information about a disease from a gene that is involved in a pathologic condition or about the pathways that the gene plays role in. Such information is often found in biomedical resources available online.

Biomedical vocabularies differ in scope and multiple vocabularies are often required at the same time, to cover concepts that are relevant to a particular biomedical application [107]. Nevertheless, manually collecting information is inefficient and susceptible to error. The identification of similar resource elements consumes time and energy and can also be a technical move that limits the rate of integration of different resources.

Patient mobility influences individual care continuity and the exchange of information are often needed for effective patient treatment. The semantic heterogeneity of the concepts in the databases often prevents the retrieval and sharing of information between biomedical databases. Biomedical resource integration was suggested as a way to promote access to various diverse resources. The integration of different biomedical concepts is gaining significance as the link between clinical medicine and biological science is growing. Speedy technological developments that have resulted in the improved analysis of complex biological systems have made cooperation and integration between the fields even more necessary [108][109]. The majority of biomedical systems have been built separately with no common framework. The database schema heterogeneity and the inconsistency of system-wide data elements are the main obstacles to the integration of data sources. A standard framework for knowledge representation could boost interactions between biological scientists and physicians, leading to better collaboration and potentially more rapid scientific discovery. The technical means for data exchange and data integration is a key issue in the management of biomedical data. Data integration can occur at many levels and semantic integration is the second to last level of integration. Ontologies form an important part of existing semantic integration approaches. These approaches can benefit from the structural properties of ontologies to map vocabularies between different sources. A lot of mapping techniques have been developed for mapping concepts between ontologies, e.g. LogMap [110], Align API

[111], and COMA++ [112]. Bioportal, the largest repository of biomedical ontologies provides mappings that associate terms between different ontologies [113]. The mappings are of different types, for example, mappings generated by the LOOM lexical matching algorithm [114], or mappings that are based on similar terms from different vocabularies in the UMLS (The Unified Medical Language System) which is a repository of many controlled biomedical vocabularies that provides a framework for mapping between these vocabularies and their relations.

Mapping biomedical vocabularies between different resources provide the ability to get from one domain to another which facilitates the integration of biomedical knowledge. Clinical data from patient care represents records of patients that harbor certain phenotypes. To cross-link between the clinical domain and molecular network domain, we need to map the concepts that represent the phenotype to their corresponding terms in biomedical databases. The mapping can be performed by comparing the similarity of the two terms. There are many techniques for vocabulary mapping. However, the most used technique is the terminological mapping approach. In this chapter, I will introduce the methods of the terminological mapping technique.

## 5.2. Materials and Methods

### 5.2.1. Terminological Mapping

The terminological mapping technique, also called lexical mapping, is based on the entities' names. It relates the concepts that have two similar names and produces a decimal value as output. This decimal value is called 'Similarity Score'. Lexical mapping techniques use similarity calculators called 'String Similarity Metrics'. They can be divided into two categories based on their calculation: character-based metrics and word-based metrics. Character-based metrics are more sensitive to character change, while word-based metrics are sensitive to word variation. Each similarity metric takes into account certain aspects or characteristics of similarity; hence ontology alignment systems need different measures to achieve greater accuracy. It should also be mentioned that string metrics differ in their output. We can define two different types: distances where the score is a positive decimal, and measures where the score is a decimal in the interval [0; 1]. Distances describe the similarity in a reversely proportional relation, where 0 means that the two strings are identical, and as the number increases, the strings are less similar to each other. For the measures, it is the opposite case. The maximum score is 1 and indicates that the strings are identical, and when the score approaches 0, the strings are less similar. In this section, I will introduce some string similarity metrics we used to lexically map biomedical vocabularies and how each one works [115] [116].

- **Jaro-Winkler** is a string metric used to calculate the edit distance between two strings that was proposed by Winkler in 1990 [117] as a variation of the Jaro distance metric [118]. The Jaro distance is the minimum number of single-character transpositions between two terms needed to transform one word into another. The calculation depends on how many matching characters the string has and the number of transpositions.

The calculation is according to the following formula:

$$Jaro(s1, s2) = \frac{1}{3} \times \left( \frac{|com(s1, s2)|}{|s1|} + \frac{|com(s1, s2)|}{|s2|} + \frac{|com(s1, s2)| - |transp(s1, s2)|}{|com(s1, s2)|} \right)$$



Where,

- $s1$  and  $s2$  are the two strings to compare
- $|s1|$  is the length of the first string
- $|s2|$  is the length of the second string
- $com(s1, s2)$  are the matching characters between  $s1$  and  $s2$
- $transp(s1, s2)$  are the characters in  $com(s1, s2)$  with different orders in  $s1$  and  $s2$  (transpositions)

The Jaro-Winkler similarity gives a boost for equal prefixes to high Jaro similarity values:

$$JaroWinkler(s1, s2) = Jaro(s1, s2) + lp(1 - Jaro(s1, s2))$$

- $l$  is the common prefix length at the beginning of the string which could be up to 4 characters maximum.
- $p$  is a constant factor of scaling for the upward adjustment of the score to be prefixed. In the work of Winkler  $p=0.1$  is the default value of this constant.

The resulting value is between 0 and 1. The distance calculated is  $1 - \text{Jaro-Winkler similarity}$  [119].

- **ISUB** [120]: This similarity metric was designed for ontology alignment. It is a character-based measure. It is based on commonalities as well as differences between two strings being compared. It is calculated according to the following formula:

$$ISUB(s1, s2) = Comm(s1, s2) - Diff(s1, s2) + JaroWinkler(s1, s2)$$

Where,

- The  $Comm(s1, s2)$  detects the longest common substring first, removes it then and constantly looks for the next longest common substring until there is no one retain. The sum of the lengths of 'i' iterations' substrings is then scaled by the length of original strings:

$$com(s1, s2) = \frac{2 \cdot \sum_i |maxComSubstring_i|}{|s1| + |s2|}$$

- $JaroWinkler(s1, s2)$  is the Jaro-Winkler similarity metric added for extra improvement.
- $Diff$  is defined as:

$$Diff(s1, s2) = \frac{uLen(s1) \times uLen(s2)}{0.6 + 0.4 \times (uLen(s1) + uLen(s2) - uLen(s1) \times uLen(s2))}$$

- $uLen$  is the distance from the original strings of the unmatched substring.
- **N-gram** [121]: This metric is a character-based similarity metric. It breaks a string or a sentence down into a set of  $n$ -grams or “shingles”. An  **$n$ -gram** is a sequence of  $n$  objects of a certain text or language sample. Depending on the application the items may be phonemes, syllables, letters, words or base pairs [122]. The measure between two strings is calculated based on the proportion of the number of  $n$ -grams that are shared between two strings and the aggregate  $n$ -grams number in both strings.

$$n\text{-gram}(X, Y) = \frac{2 \times |n\text{-grams}(X) \cap n\text{-grams}(Y)|}{|n\text{-grams}(X)| + |n\text{-grams}(Y)|}$$

Where,

- $n\text{-grams}(X)$  is the set of multiple letter sequence of  $n$ -grams in  $X$ .
- $n\text{-grams}(Y)$  is the set of multiple letter sequence of  $n$ -grams in  $Y$ .
- **Levenshtein** [123]: It is an edit distance metric. It represents the minimum number of single-character deletions, insertions or replacements, needed for converting one word to another.

$$lev_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1, j) + 1 \\ lev_{a,b}(i, j-1) + 1 \\ lev_{a,b}(i-1, j-1) + 1_{(a_i \neq b_i)} \end{cases} & \text{otherwise.} \end{cases}$$

where ,

- $1_{(a_i \neq b_i)}$  is the function of the indicator which is equals to 0 if  $a_i = b_i$  and 1 otherwise
  - $lev_{a,b}(i,j)$  is the distance between the first characters  $i$  of  $a$  and the first character  $j$  of  $b$ .  $i$  and  $j$  are 1-based indices.
- **Normalized Levenshtein** [124]: It is calculated by dividing the distance of Levenshtein by the length by the longest string. It results in a non-metric value between  $[0, 1]$ . Then,  $1 - \text{normalized distance}$ , is the calculated similarity [119].
  - **Damerau Levenshtein** [125][126]: This metric calculates the minimum number of operations needed for the conversion of one string into another. An operation is based on the single character addition, deletion or replacement, or a transposition of two adjacent characters [119].

$$d_{a,b}(i,j) = \min \begin{cases} 0 & \text{if } i = j = 0 \\ d_{a,b}(i-1,j) + 1 & \text{if } i > 0 \\ d_{a,b}(i,j-1) + 1 & \text{if } j > 0 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)} & \text{if } i,j > 0 \\ d_{a,b}(i-2,j-2) + 1 & \text{if } i,j > 1 \text{ and } a[i] = b[j-1] \text{ and } a[i-1] = b[j] \end{cases}$$

where ,

- $1_{(a_i \neq b_i)}$  is the indicator function equal to 0 when  $a_i = b_i$  and equal to 1 otherwise.
- **Longest Common Subsequence (LCS)** [127]: It is a distance that works by identifying the common longest subsequence present in two or more sequences. This varies from the issues of identifying common substrings: as opposed to substrings are not needed to take sequential positions in the initial sequences [119]. It is calculated according to the following formula where  $n$  and  $m$  are the respective lengths of  $s1$  and  $s2$ :

$$LCSub(s1,s2) = n + m - 2|LCS(s1,s2)|$$

If no replacement exists and only insertion and removal is permitted, or if the substitute cost is twice the insertion or deletion cost, the LCS distance is equivalent to the Levenshtein distance [128].

- **Pre-computed Cosine** [129]: This is a normalized distance where the similarity is the cosine of the angle between two vector representations, and is computed as:

$$PreCos(v1, v2) = \frac{v1 \cdot v2}{|v1| * |v2|}$$

- **Jaccard Index** [130]: This metric is a word-based distance. It is defined as the two-word set intersection divided by the union of them. If A and B represent respectively the sets of words of s1 and s2, then

$$Jaccard(s1, s2) = \frac{|A \cap B|}{|A \cup B|}$$

- **Sorensen-Dice Coefficient** [131]: This metric is similar to the Jaccard index metric, but it is calculated according to the following formula:

$$Dice(s1, s2) = 2 * \frac{v1 \cap v2}{|v1| + |v2|}$$

- **String Kernel**: is a metric that calculates a similarity between two words, based on counts of common subsequences of characters by using a kernel function. It has been presented for text classification by *Lodhi et al.* [132]. String kernels can be understood as functions that calculate similarity between string pairs: the more similar are two strings a and b, the greater the value of a string kernel  $K(a, b)$  will be [133].

### **5.2.2. International Classification of Diseases (ICD)**

Clinical data from patient care represents records of patients that harbor certain phenotypes. The International Disease Classification (ICD) [10] system is the diagnostic classification standard that provides disease codes used in clinical records to identify these phenotypes. It is published, copyrighted, and updated regularly by the World Health Organization (WHO) [10].

ICD-10 codes are alphanumeric codes used by doctors, health insurance companies, and public health organizations all over the world to report diseases and health conditions. Each disease, disorder, injury, infection, or symptom is assigned by its own ICD-10 code. The codes are used to process health insurance claims, monitor disease epidemics and compile global mortality statistics [134]. The codes can be used to mine the administrative data associated with clinical records in order to apply some data analysis on clearly identifiable phenotypes. ICD defines the universe of clinical terms described in a hierarchical manner that allows health information to be easily stored, retrieved, and analyzed. It is used to share and compare health information between regions and countries, and to compare data in the same location over various time periods. The latest version (2016) of ICD-10 structure is available online by the World Health Organization (WHO) [10] (Figure 17) and by BioPortal [135], but it cannot be exported. In our work, we used the ICD10 Ontology which is formalized in OWL-DL of the 10th edition of the International Classification of Diseases published in 2004 by the World Health Organization (WHO) [136]. We downloaded the ontology and exploited the ontology structure.


- 
- ▼ ICD-10 Version:2019 
- ▶ I Certain infectious and parasitic diseases
  - ▶ II Neoplasms
  - ▶ III Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
  - ▶ IV Endocrine, nutritional and metabolic diseases
  - ▶ V Mental and behavioural disorders
  - ▶ VI Diseases of the nervous system
  - ▶ VII Diseases of the eye and adnexa
  - ▶ VIII Diseases of the ear and mastoid process
  - ▶ IX Diseases of the circulatory system
  - ▶ X Diseases of the respiratory system
  - ▶ XI Diseases of the digestive system
  - ▶ XII Diseases of the skin and subcutaneous tissue
  - ▶ XIII Diseases of the musculoskeletal system and connective tissue
  - ▶ XIV Diseases of the genitourinary system
  - ▶ XV Pregnancy, childbirth and the puerperium
  - ▶ XVI Certain conditions originating in the perinatal period
  - ▶ XVII Congenital malformations, deformations and chromosomal abnormalities
  - ▶ XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
  - ▶ XIX Injury, poisoning and certain other consequences of external causes
  - ▶ XX External causes of morbidity and mortality
  - ▶ XXI Factors influencing health status and contact with health services
  - ▶ XXII Codes for special purposes

Figure 17. A screenshot for the hierarchy of the International Classification of Diseases and Related Health Problems 10th Revision (Source <https://icd.who.int/browse10/2019/en>)

### 5.2.3. Medical Subject Headings (MeSH®)

The Medical Subject Headings (MeSH®) thesaurus is a controlled vocabulary established by the US National Library of Medicine (NLM) that provides hierarchically-structured concepts [137]. It is commonly used for indexing articles for the MEDLINE®/PubMed [138] database and other databases. MEDLINE® is the largest online database containing nearly all articles published in the field of biomedicine. With PubMed, these published articles can be accessed on the web for free [139]. The indexing of MEDLINE® citations by MeSH® terminology

allows a reliable means to identify citations, even if different terms for the same concept are used by authors [140].

The MeSH® vocabulary can be explored using the online MeSH® browser [141]. Every MeSH® term in the vocabulary represents a biomedical concept being used in literature [8]. MeSH® terms help to focus the search for more relevant articles. Each term in the thesaurus is represented by a MeSH® heading term, a unique ID, and entry terms (Figure 18). The entry terms, also known as "cross-references", are synonyms, alternative forms, and other closely related terms in a certain MeSH® record, usually used interchangeably with the preferred descriptor term for indexing or retrieval purposes [142].

**MeSH Heading:** Parkinson Disease

**Entry Term:** Idiopathic Parkinson Disease  
**Entry Term:** Idiopathic Parkinson's Disease  
**Entry Term:** Lewy Body Parkinson Disease  
**Entry Term:** Lewy Body Parkinson's Disease  
**Entry Term:** Paralysis Agitans  
**Entry Term:** Parkinson Disease, Idiopathic  
**Entry Term:** Parkinson's Disease  
**Entry Term:** Parkinson's Disease, Idiopathic  
**Entry Term:** Parkinson's Disease, Lewy Body  
**Entry Term:** Primary Parkinsonism

**Unique ID:** D010300

Figure 18. 'Parkinson Disease' term in MeSH® vocabulary.

#### 5.2.4. National Cancer Institute Thesaurus (NCIT)

The National Cancer Institute Thesaurus (NCIT) [143] is a reference terminology that extensively covers the cancer field, including diseases, findings, and abnormalities related to cancer. This resource displays definitions and linked information related to over 10000 cancers and 8000 single agents and combination therapies. The terminology can be explored online [143] and can be downloaded as an OWL or OBO ontology from several sources such as OBO Foundry [144] and BioPortal [86] (Figure 19). The National Cancer Institute's Thesaurus (NCIT) was developed to provide a standardized language for experts in various subdomains of oncology. This is intended for annotation purposes so that data and information derived from

these different sub-domains are integrated and thus more efficient cross-domain inferences can be supported [145].

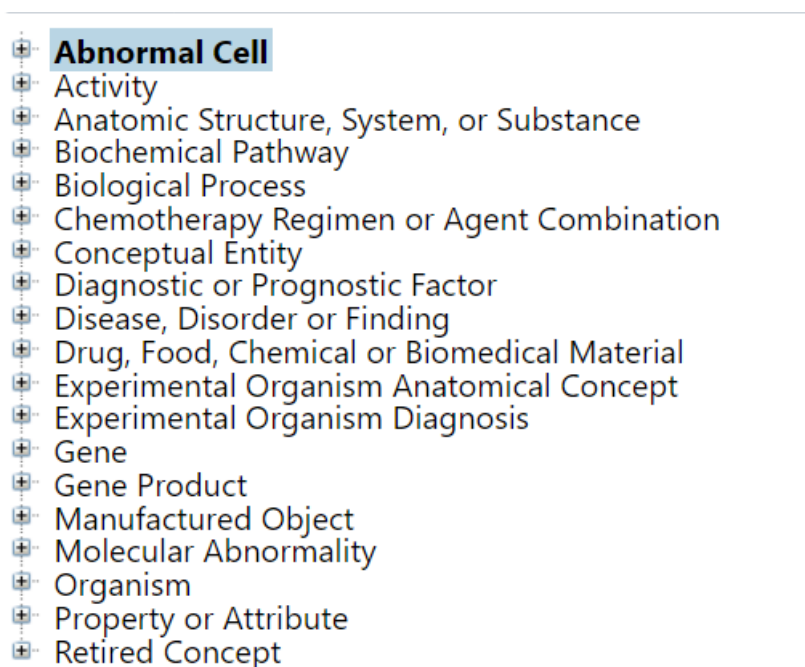


Figure 19. The NCIT ontology top-level structure in BioPortal.

### 5.2.5. HumanPSD (Human Proteome Survey Database)

The HumanPSD [146] is a database of diseases, signaling pathways, biomarkers, drugs, and drug targets. It is a trademark of QIAGEN, distributed by geneXplain GmbH [80]. It is a rich information resource of over 2,500 diseases including information about clinical trials ongoing and ended around the globe. All information in HumanPSD is manually curated from scientific literature or extracted from clinical trial catalogs. HumanPSD has a wealth of molecular function information that can help to uncover biologically relevant connections between sets of genes and proteins to disease and drugs. HumanPSD contains TRANSPATH [147][39], a database of mammalian signaling and metabolic pathways and networks with manually curated information. TRANSPATH reports individual reactions with all experimental information in a precisely mechanistic manner including all reaction partners as reported in the publication. This helps to build the most realistic model of the intracellular pathways acting in various diseases. HumanPSD with the TRANSPATH database allows the connection of signaling pathways with targets, drugs, and clinical trials.



## 5.3. Results

### 5.3.1. Lexical Mapping Module Implementation

We developed a lexical mapping module that integrates all the similarity metrics mentioned before to evaluate the lexical similarity between two concept names. The module is implemented in Java and the similarity metrics are implemented and wrapped into Java libraries.

Before the lexical mapping calculation starts, there is a pre-processing phase to improve the calculation. Some concepts can have minor character differentiation because they might have been written by two different groups of scientists, which induces a necessary variation in the naming standards. For instance, the underscore symbol ( \_ ) can exist in some concepts to separate words of a concept, while it's replaced with a space in another concept. Other preprocessing techniques can be applied such as lowercasing, removing stop words, replacing British/US spelling letters, and lemmatization (reducing words into their lemma) (Figure 20). The differences between these characters can lower the scores of two exact or similar concepts. The pre-processing phase eliminates naming standards' variation in both concepts.

## Preprocessing

### Lowercasing

- Refractory **A**nemia with **E**xcess of **B**lasts
- refractory anemia with excess of blasts

### Removing stop words (of, the, in, for...)

- Muscle Disorders                      Disorders **of** muscles

### Replacing British/US letters (ae,e) (oe,e)...

- Lymphoid Leukaemia                      Lymphoid Leukemia
- Oestrogen                                      Estrogen

### Lemmatization

- Monoclonal Gammopathies                      Monoclonal Gammopathy

Figure 20. The preprocessing techniques.

The lexical calculation is computed using the String Similarity Metrics, and since every metric solves one problem, we chose the metrics introduced before and combined them to produce one score. Every metric is assigned by weight to give importance to a metric that can solve a particular problem according to the purpose. Moreover, in some cases, the concepts to compare can be large and produce a long list of mapping results and the valuable results are above a certain score value, therefore a threshold is used. The weights and the thresholds can be modified by the user. The score of every metric is calculated according to the following formula:

$$\mathit{score} = \mathit{sim}_k (s1, s2) \times w$$

Let  $\omega$  be the set containing the weights for all string metrics, and  $\mathit{sim}$  is the set containing the scores of all the metrics. The final score is called the *aggregated similarity score* and is calculated using the following formula:

$$\mathit{sim}_{agg}(s1, s2) = \frac{\sum_{k=1..n} \omega_k \cdot \mathit{adj}(\mathit{sim}_k(s1, s2))}{\sum_{k=1..n} \omega_k}$$

Where  $\mathit{adj}(\mathit{score})$  is a function used to unify the forms of the scores, when using different types of metrics, like distances and measures.

The lexical mapping is relatively simple and works as follows: for each concept in one source, the similarity score is calculated with every concept in another source, and the result is stored in a new instance of a class called “*Candidate*” containing both concepts and the similarity score. The candidate for one concept of the first source having the highest score is chosen. However, for example, if a source has 1008 concepts, and the other source has 9998 classes. Consequently, around  $10^7$  candidates are calculated.

The concepts that have similar names will have higher scores than concepts with different names. The resulting scores are positive decimals less than ‘1.0’. A score of ‘1.0’ means the concepts have exact matching names.

The module takes as input two lists of terms with their codes, each list corresponds to one source. A score is calculated for each metric, and then the aggregated score is calculated (Figure 21).

The output is a table that includes the terms, the codes, and the score values. The module can be used as a standalone Java tool or it can be integrated into other applications such as ontology tools to perform ontology alignment (Figure 21).

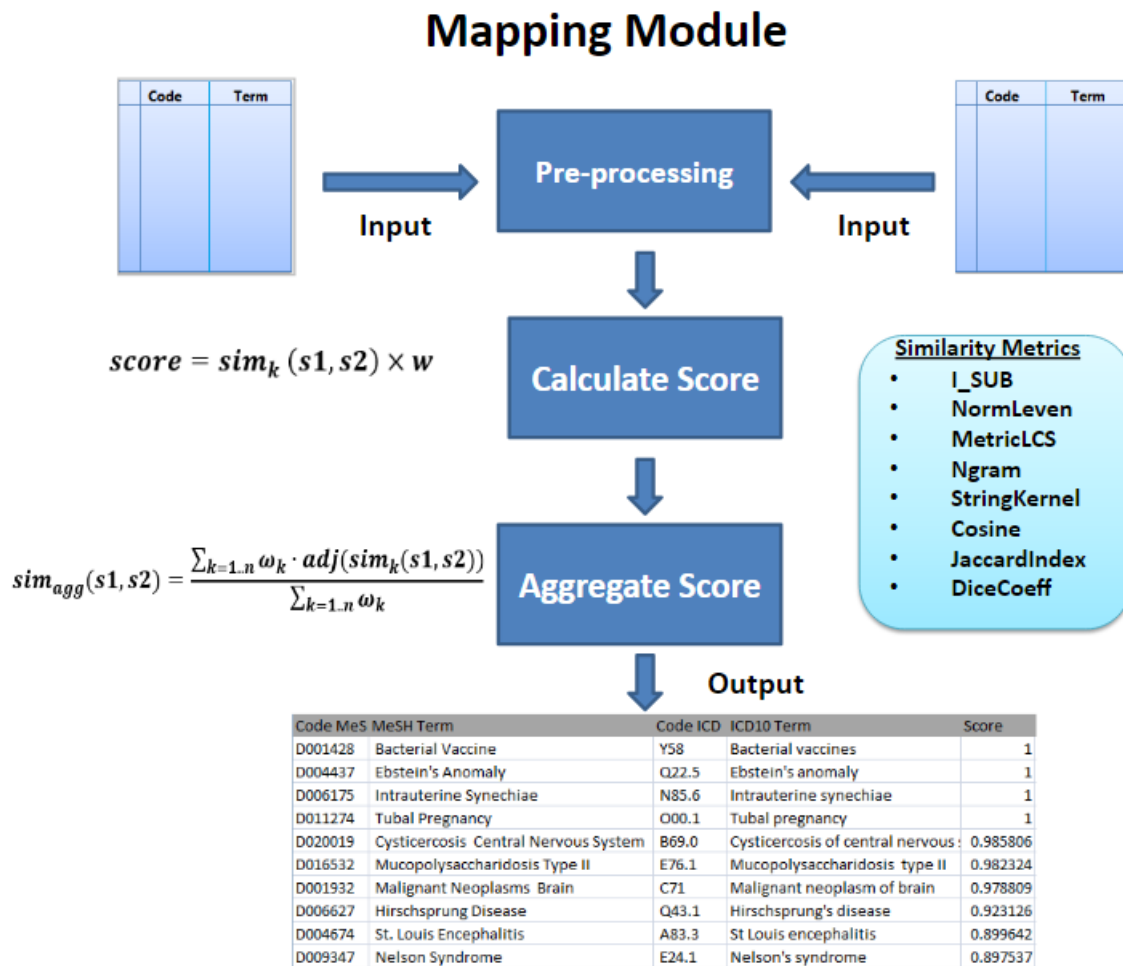


Figure 21. The lexical mapping module architecture.

### 5.3.2. Link Clinical Data to Biomedical Data

More efficient data use in Systems Medicine can be provided by integrating patient clinical and genomics data with pathway knowledge. It has become evident that genes do not function alone inside a cell. They interact with each other in a way to create pathways or complexes to conduct biological functions and contribute to the etiology of complex diseases such as Cancer. In certain disorders, the disease candidate genes have been shown to be functionally linked as biological pathways or protein complexes [148]. Combining molecular biology and genetics has made it much easier to classify candidate genes associated with human disease. Therefore,

discovering relationships among human diseases and biological pathways, through overlapping genes, could provide new insights into the etiology of diseases.

The need for molecular and clinical information within a unified framework has long figured as a grand challenge in biomedical informatics. Discovering relationships between human diseases and biological pathways, based on genes, gives new insights into disease etiology. Mapping of biomedical vocabularies in different resources provides the ability to get from one domain to another which facilitates the integration of biomedical knowledge. To cross-link between the clinical domain and molecular network domain, we need to map the concepts that represent phenotypes to their corresponding concepts in biomedical databases. The link provides the ability to explore molecular information through clinical data.

#### **5.3.2.1. Link ICD Concepts to NCIT Concepts**

As a first step to explore molecular data and pathways that affect certain clinical phenotypes is to have a link from the clinical data to molecular knowledge frameworks. Our work focuses mainly on neoplasms as phenotypes. ICD-10 codes from the clinical data represent disease names that are assigned to every patient who has a certain disease. The ICD hierarchy consists of multiple chapters, based on the subject of the ICD codes each chapter contains. The neoplasm chapter provides codes for benign and malignant neoplasms. NCIT a crucial cancer-related reference terminology. To integrate clinical phenotype concepts with molecular data space, a cross-link can be provided by mapping the ICD concepts to disease concepts in the NCIT. To limit our work to neoplasms, we used only the sub-concepts under the “Neoplasms” chapter in the ontology hierarchy of the ICD classification (Figure 22), and the disease concepts that have “neoplastic process” as a semantic type which is a class property in the NCIT ontology.

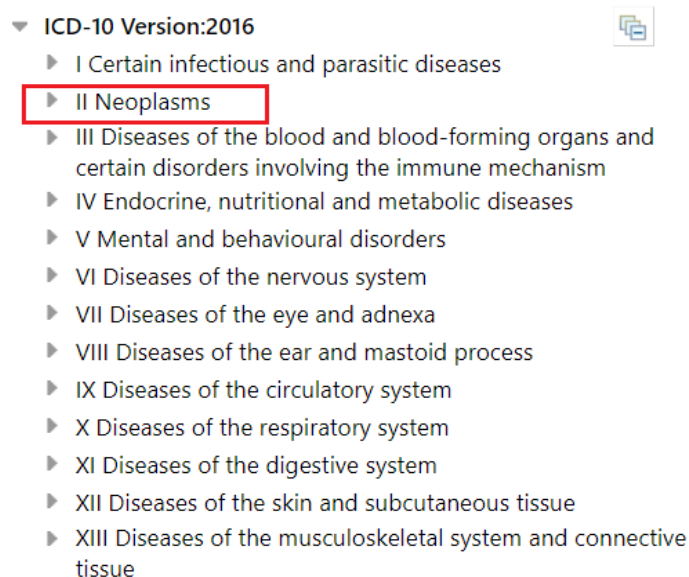

- 
- ▼ ICD-10 Version:2016 
  - ▶ I Certain infectious and parasitic diseases
  - ▶ II Neoplasms
  - ▶ III Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
  - ▶ IV Endocrine, nutritional and metabolic diseases
  - ▶ V Mental and behavioural disorders
  - ▶ VI Diseases of the nervous system
  - ▶ VII Diseases of the eye and adnexa
  - ▶ VIII Diseases of the ear and mastoid process
  - ▶ IX Diseases of the circulatory system
  - ▶ X Diseases of the respiratory system
  - ▶ XI Diseases of the digestive system
  - ▶ XII Diseases of the skin and subcutaneous tissue
  - ▶ XIII Diseases of the musculoskeletal system and connective tissue

Figure 22. The “Neoplasms” chapter in the ICD 10 classification hierarchy.

We applied a lexical automatic mapping to compare every ICD term with a disease concept (preferred term) and its synonyms in the NCIT. For ICD-10, to extract only the terms that are classified as neoplasms, we need to get access to the hierarchy of the classification. Under certain circumstances is ICD only published by the WHO itself, so that the classification is not available for public use. However, one can get it anyway from the WHO by reading through their contract. Therefore, we used the ICD-10 ontology from Data & Knowledge Management (DKM). It is a formalized ICD10 Ontology in OWL-DL of the 10th edition of the International Classification of Diseases, published in 2004 by the World Health Organization (WHO) [149]. We used the OBA service to get access to the ontology structure and to extract the concepts that are at two levels in the hierarchy. The first level is the lowest level that includes the most concrete concepts, e.g. “Malignant neoplasm: Anterior floor of mouth”, and we consider it as “low level”. The second level is the next higher level than the lowest one, and we consider it as “high level” (Figure 23).

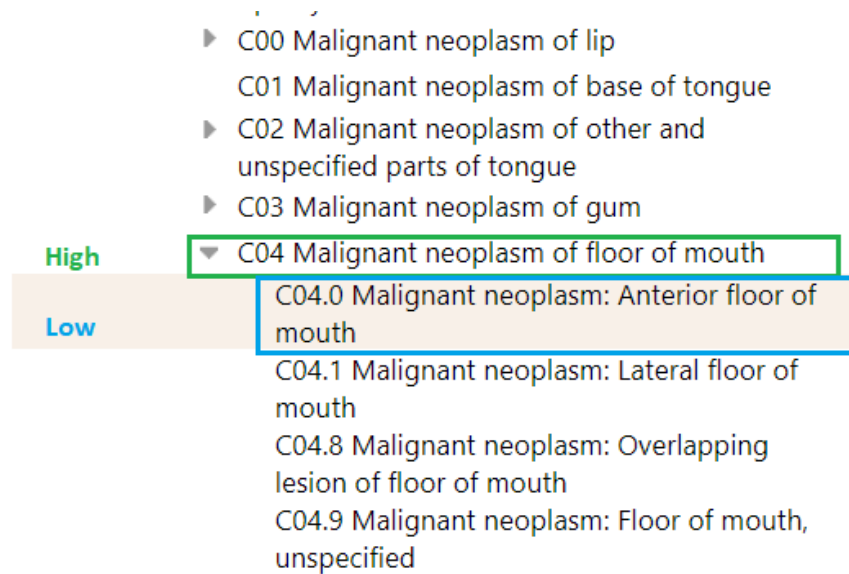


Figure 23. Part of the ICD hierarchy that shows the levels we used.

To limit the data in NCIT to neoplasms, we used the OBA service as well to get access to the ontology and to extract the terms that have the “neoplastic process” semantic type property. Figure 24 shows an example of the “Malignant\_Breast\_Neoplasm” class with the code “C9335” in its properties in the ontology while it is loaded in the OBA service. The “P366” property is the class label and the “P90” is the class synonym which is more than one. The semantic type is represented by the “P106” property in the ontology. Using functions that were specifically developed for the NCIT ontology, we extracted the concepts that have the “Neoplastic Process” property with their labels and their synonyms.

<http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C9335>

## Properties

P90	<input type="checkbox"/> Malignant Breast Tumor
P366	<input type="checkbox"/> Malignant_Breast_Neoplasm
P90	<input type="checkbox"/> Malignant Tumor of Breast
P90	<input type="checkbox"/> Malignant Neoplasm of Breast
P90	<input type="checkbox"/> Malignant Tumor of the Breast
P97	<input type="checkbox"/> A primary or metastatic malignant neoplasm involving the breast. The vast majority of cases are carcinomas arising from
P90	<input type="checkbox"/> Malignant Breast Neoplasm
NHC0	<input type="checkbox"/> C9335
P90	<input type="checkbox"/> Malignant Neoplasm of the Breast
P207	<input type="checkbox"/> C0006142
P108	<input type="checkbox"/> Malignant Breast Neoplasm
P363	<input type="checkbox"/> Malignant
label	<input type="checkbox"/> Malignant Breast Neoplasm
P106	<input type="checkbox"/> Neoplastic Process

Figure 24. The “Malignant Breast Neoplasm” class in the NCIT ontology loaded in the OBA service. P106 is the semantic type property.

Using all the filtered data, we applied a mapping between the ICD 10 terms and NCIT terms that works by matching the terms using the lexical matching approach we developed.

We applied the mapping at first, at the “high level” (parents) from the ICD under the neoplasms chapter, and by adjusting our methods we got around 43% of the ICD terms that can be exactly mapped with a score equal to 1. Then the same methods have been applied to the terms at the “low level” (children) and as a result, we got around 30% of ICD terms that can be exactly mapped to the NCIT terms. The mapping works by comparing each ICD term to a preferred term from the NCIT and its synonym. If the matching between an ICD term and one of the synonyms produces a score equal to 1, then the preferred term is used to be mapped to the ICD term (Figure 25).

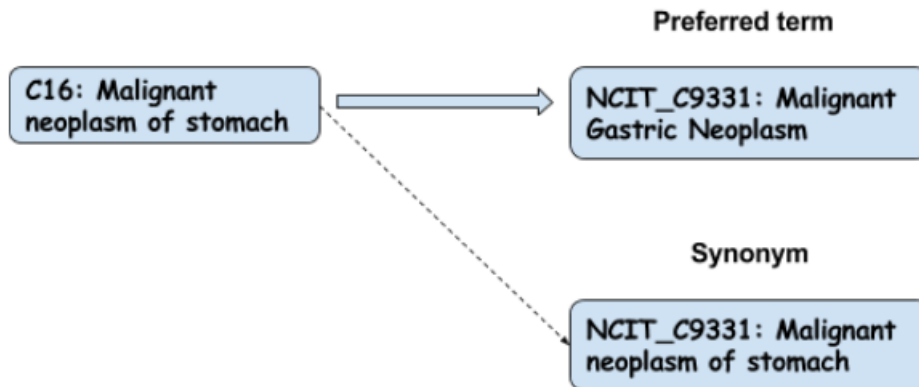


Figure 25. An exact lexical mapping that is based on the string matching of an ICD term that matches a term’s synonym in the NCIT.

The more the ICD terms are concrete, the less efficient the lexical mapping is. To improve our mappings and try to map terms from the ICD as much as possible, human intervention and other mapping strategies are needed. Since the lexical mapping can give exact mappings for some terms, we can use it as one mapping strategy and combine other strategies that can help to improve the results. Many of the concrete terms from the lowest level in the ICD cannot have exact mapping, so we can use the mappings of their parents as alternatives since the children describe part of their parents. For the highest level in the ICD, the remaining terms (other than the 43%) can be mapped manually to have valid mappings and to use them for structural mapping. Once we have these mappings, we can apply a semi-automatic mapping. The semi-automatic mapping was applied to the terms at the “lowest level”. For these terms, since their parents can already have mappings whether automatically or manually, they can be mapped to the same terms that their parents are mapped to by using a structural mapping strategy. In this case, we will be able to produce more mappings.

Our semi-automatic approach is restricted to the ICD and the NCIT as input, but it can be adjusted to be applied to other data. It is applied to one level in the ICD, the lowest one that includes the most concrete terms. It starts by applying and comparing the ICD terms and the preferred terms that belong to the “neoplastic process” in the NCIT using a lexical mapping. And then the same strategy is applied to the synonyms of the same terms in the NCIT, but by using the preferred terms as mapping terms. The results produced from both steps are combined and the mappings which have a score equal to 1 are kept. For the results below of score = 1, we perform a manual mapping to match the terms at the highest level of the ICD with NCIT terms. Then a structural mapping is applied, and every term will be mapped to the term that its



parent is mapped to (Figure 26). In the end, the results are divided into two sections, the first one that includes the manual mapping of the highest level, and the second one of the semi-automatic mapping applied to the lowest level.

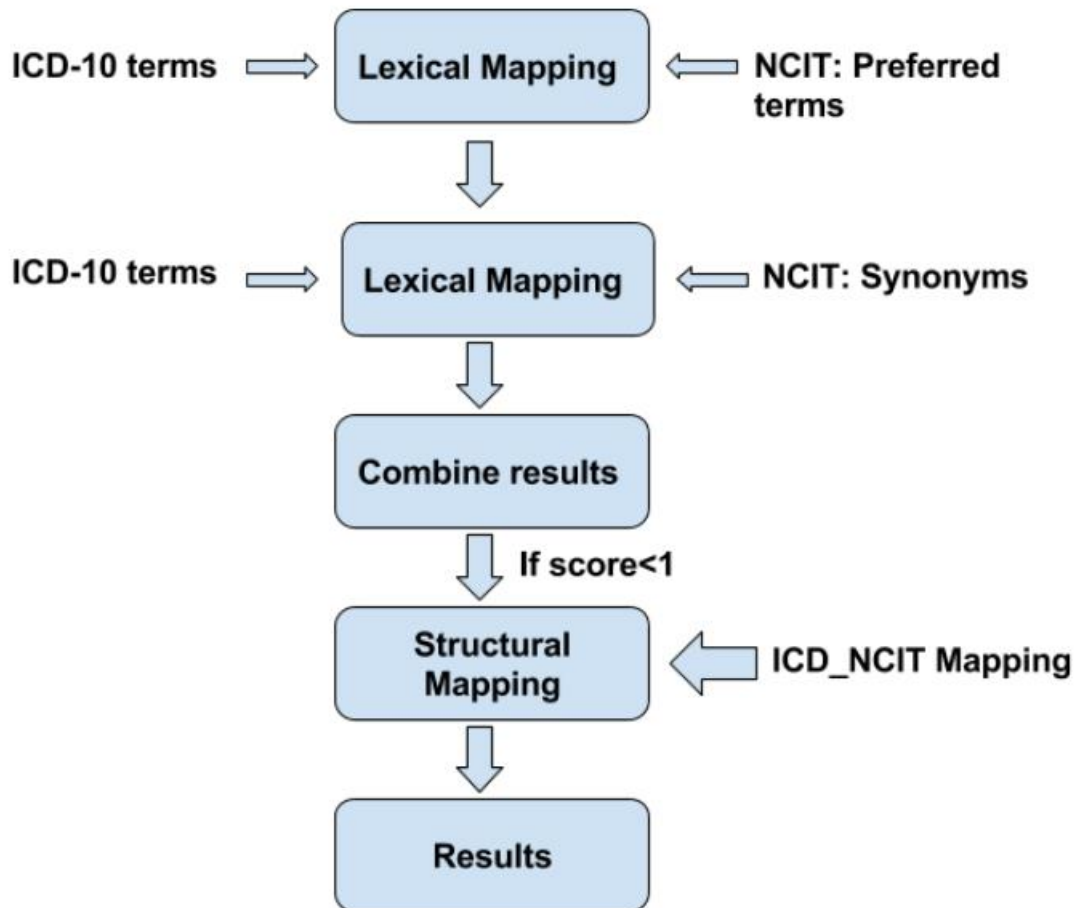


Figure 26. Semi-automatic mapping module.

The results we produced are available on the web page of the project that this work is part of (<http://mypathsem.bioinf.med.uni-goettingen.de/research/workpackage-4>). Figure 27 gives a glimpse of the results of the ICD mapped terms at the “low level”. They are in a tabular format and include the ICD terms, their codes, their matched NCIT terms, the codes of the NCIT terms, and the mapping technique name.

D12.1	Benign neoplasm of appendix	C4773	Benign Appendix Neoplasm	Lex
D12.0	Benign neoplasm of caecum	C4772	Benign Cecum Neoplasm	Lex
C03.9	Malignant neoplasm of gum unspecified	C9317	Malignant Gingival Neoplasm	Syn
D12.6	Benign neoplasm of colon unspecified	C2894	Benign Colon Neoplasm	Lex
D12.8	Benign neoplasm of rectum	C4774	Benign Rectal Neoplasm	Lex
C14.0	Malignant neoplasm of pharynx unspecified	C7545	Malignant Pharyngeal Neoplasm	Syn
C18.7	Malignant neoplasm of sigmoid colon	C9242	Malignant Colon Neoplasm	Stru
C16.4	Malignant neoplasm of pylorus	C3552	Malignant Gastric Neoplasm	Stru
C71.5	Malignant neoplasm of cerebral ventricle	C4577	Malignant Brain Neoplasm	Stru

Figure 27. A glimpse of the ICD-NCIT mappings results.

### 5.3.2.2. NCIT Plugin in the OBA service

Pathway information is not directly connected to disease information in the NCIT ontology. A gene in the NCIT has the “Gene\_Associated\_With\_Disease” relation which is used to assert a link between a gene and a disease when the association is considered to have clinical relevance. This relation allows us to get the diseases that are associated with genes. Moreover, exploration of the gene-interaction pathways involved in disease is a rapidly growing area of research in cancer. KEGG [99] and Biocarta [150] are two well-known maintainers of the pathway information in the NCIT. The pathways that play a role in such diseases can be supported by a role relationship that links the disease to the gene hierarchy, and from there to the pathway hierarchy (Figure 28).

A gene has the “Gene\_Is\_Element\_In\_Pathway” relation that relates it to a biochemical pathway in which its encoded gene product participates. Therefore, the gene entities can be used to link diseases to pathways.

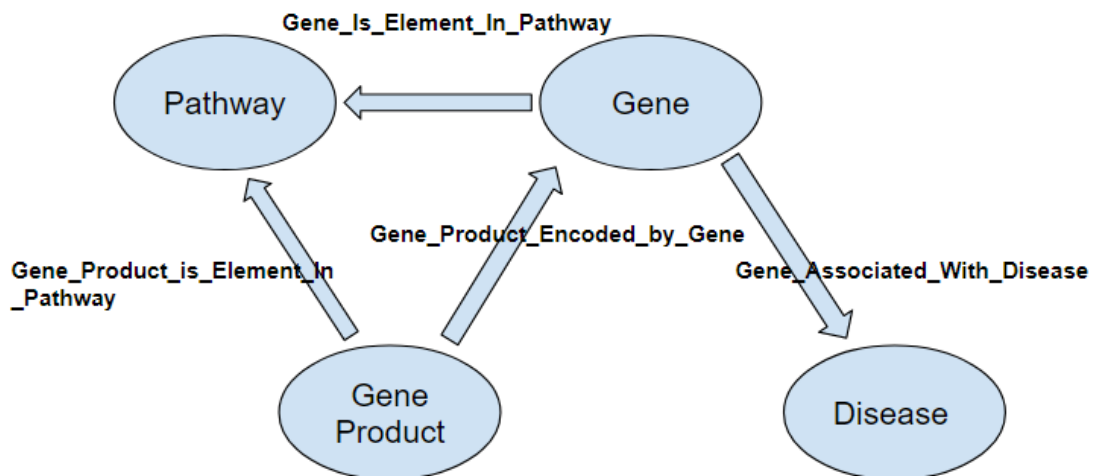


Figure 28. Part of the semantic model in the NCIT.

We developed a plugin for the NCIT ontology in the OBA service to implement specific functions that can allow us to model disease pathways using the ontology structure and the relations between its entities. To explore pathways associated with a certain disease, we need to have a query that starts with a disease term. To get genes associated with diseases, we extracted all the gene entities with their names (which correspond to gene symbols) from the ontology and stored them as a list in a database. For each gene and by using the “Gene\_Associated\_With\_Disease” relation, we implemented the “GenesAssociatedWithDisease” function to get a list of diseases associated with a certain gene using a gene name (gene symbol) as query input. Figure 29 shows an example of the disease classes that are returned using the “GenesAssociatedWithDisease” query we developed, and they are related to the ‘CHEK2’ gene in the OBA service while the NCIT ontology is loaded. The disease classes can be queried by using a string query with a gene name in the URL. Figure 30 shows the labels of the returned disease classes in the OBA console.

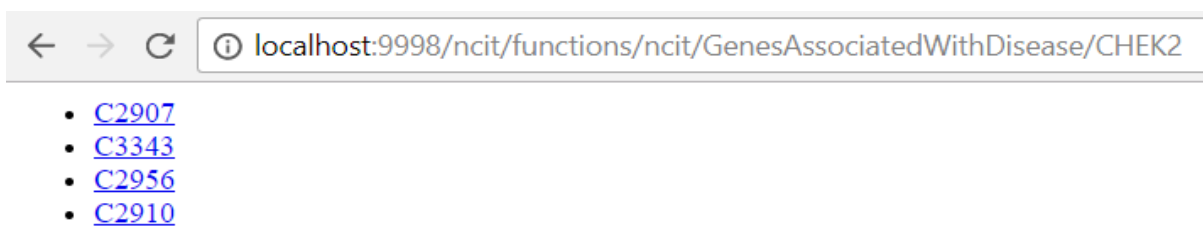


Figure 29. A screenshot of the disease classes associated with CHEK2 gene in the OBA service.

```
INFO-The code of the gene CHEK2 is C40965.  
INFO-restrictions 9  
INFO-Disease restrictions: 4  
INFO-The diseases associated with gene are:  
Brain Neoplasm  
Prostate Neoplasm  
Colorectal Neoplasm  
Breast Neoplasm
```

Figure 30. The diseases associated with the CHEK2 gene in the console.

The function was applied to every gene extracted from the ontology and the results were stored as gene-disease associations in a database with a many-to-many relation.

Using these associations, we implemented the DiseaseHasAssociatedGenes(disease) function to get genes associated with a certain disease by using the disease name as input (Figure 31).

(<http://localhost:9998/ncit/functions/ncit/DiseaseHasAssociatedGenes/breast%20neoplasm>)

```
INFO-The code of the disease code found C2910!  
INFO-The gene codes related to breast neoplasm are 4  
INFO-C20796--ERGIC3 Gene  
INFO-C20797--PRICKLE4 Gene  
INFO-C20718--DDR1 Gene  
INFO-C40965--CHEK2 Gene  
INFO-Done!
```

Figure 31. The genes associated with Breast neoplasm.

Using the “Gene\_Is\_Element\_In\_Pathway” relation that relates a gene with a pathway in the ontology, a function was implemented to get the pathways that a certain gene is an element in. The allele or the gene product of the gene is related to the pathways, not the gene. Hence, an association between the gene and the associated pathways must be done. Using this association and the “Gene\_Is\_Element\_In\_Pathway” relation, we implemented the “GeneInPathway” function to get the pathways that a gene is an element in using the gene name (gene symbol) as input (Figure 32)

(<http://localhost:9998/ncit/functions/ncit/GeneInPathway/PPM1D>)

```
INFO-The code of the gene PPM1D is C88180.  
INFO-restrictions 2  
INFO-Pathways restrictions: 2  
INFO-The pathways that the gene PPM1D is element in are:  
p53 Signaling Pathway KEGG  
p53 Pathway for Transcriptional Regulation
```

Figure 32. The pathways that the gene PPM1D is element in.

We used this function to implement another function that takes a list of genes as input and gets the pathways that all genes in this list are elements in. The gene list should be provided as a CSV file. The other way around can be also done, which is getting the genes that are elements in a certain pathway given by the user.

Role relationships support the retrieval of information on a pathway of interest, the identification of a gene of interest in that pathway, and the discovery of any cancers known to be linked to that gene or its products. We used some other ways around to retrieve pathway information and to get to pathways from diseases (Figure 33). We stored data from NCIT in a MySQL database to retrieve information that was used differently to complete the functionality of the implemented functions. The database is connected to the NCIT plugin in the OBA service.

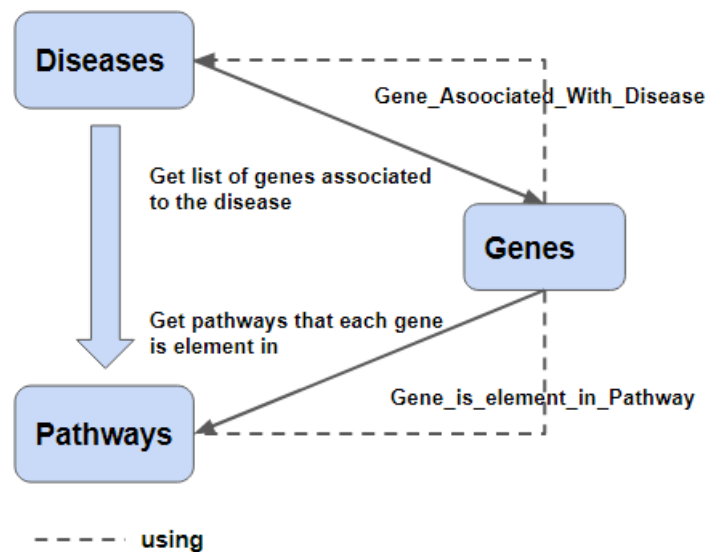


Figure 33. Extracting the disease/pathways associations through genes.

Using all the implemented functions, we were able to develop the “DiseaseHasAssociatedpathways(disease)” function that can provide the possible pathways that have associations with a certain disease based on genes and by using a disease label as input).

Example query:

<http://localhost:9998/ncit/functions/ncit/DiseaseHasAssociatedpathways/breast%20carcinoma>

Due to some missing implemented functions in the OBA functionality, we were not able to extract all the existing pathways through this relation. Once this problem is solved, we can get all the other possible pathways. Figure 34 shows an example where the “DiseaseHasAssociatedpathways(disease)” function is applied to “breast carcinoma” and returns two associated pathways.

```

INFO-The pathways related to the disease breast carcinoma
INFO-Pathway : C91495--p53 Signaling Pathway KEGG
INFO-Pathway : C91571--p53 Pathway for Transcriptional Regulation
INFO-Done!

```

Figure 34. The pathways associated with Breast Carcinoma.

The OBA service with the developed plugin, the connected database, and all the libraries and other dependencies were packaged up into a Docker container. Docker [151][152] is a tool designed to simplify the process of creating, deploying and running applications by using containers. It provides a lightweight environment to run an application code. A container makes it easier to deploy the application as one package and to run it on any Linux machine regardless of any customized settings that machine might have. The MySQL server uses the port 3306 and the OBA server runs on port 9998. Docker maps the ports to the container port 80 and then it exposes it to the host port 8080 to run the container (Figure 35). The Docker container of this application is available on our project server.

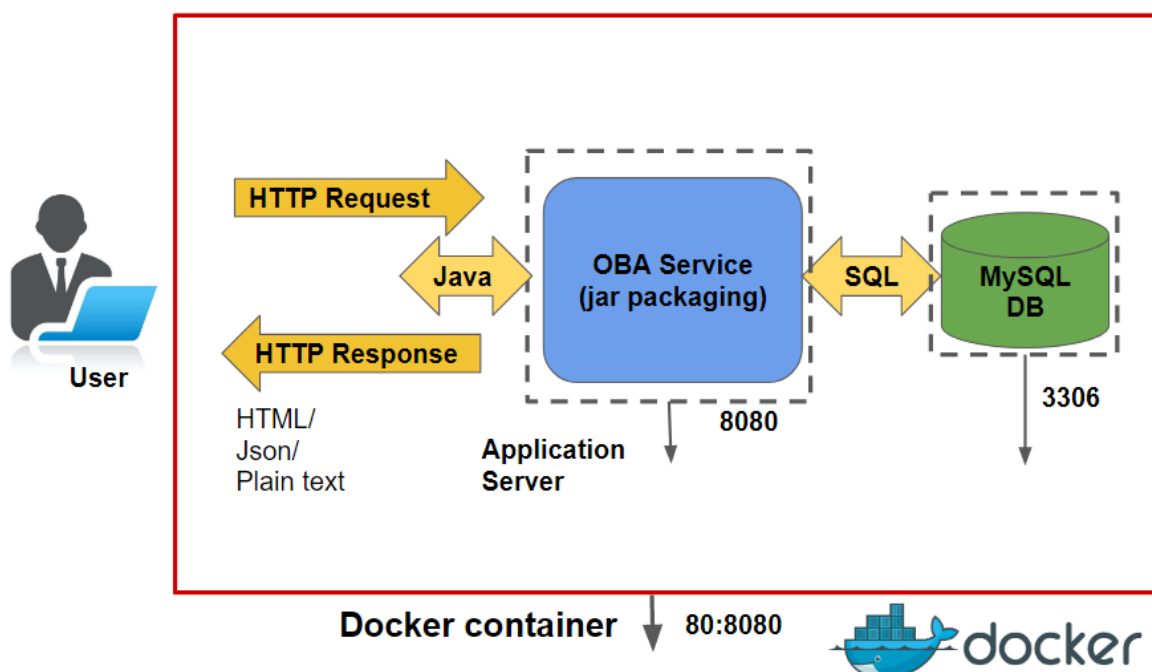


Figure 35. The architecture of the packaged application in Docker.

### 5.3.2.3. Link ICD Concepts to MeSH® Concepts

To provide more efficient data use in Systems Medicine, HumanPSD was another option to enable further medical and pharmaceutical insights. A link from patient clinical codes (ICD) to molecular and pathway information, and clinical trials in HumanPSD, can be enabled through diseases. Diseases reported in HumanPSD are represented by MeSH® identifiers and labels. A mapping between ICD terms and MeSH® terms can facilitate the use of such information in decision-making systems. Resources like Bioportal and UMLS provide mappings between ICD and MeSH® that are based on different strategies. Bioportal mappings [113] are relations between two or more concepts in different ontologies that represent a degree of similarity. We got 2,697 ICD-MeSH® mappings from Bioportal by using the Rest API service. These mappings represent around 20% of the ICD terms.

In UMLS, a MeSH® term and an ICD 10 term share the same concept under the same UMLS code. Using this strategy, we got the mappings between ICD and MeSH® which represent 10% of the ICD terms. Providing as many as possible mappings between resources can provide more connectivity between them. Therefore, we applied the mapping strategies we used before to map ICD terms to MeSH® terms. We started with the same strategy we did before, which is limiting the ICD terms to the terms that belong to the “Neoplasms” category and dividing them into two levels “high level” and “low level”. By applying the lexical mapping approach, we got around 10% of the ICD terms at the “high level” and around 3% at the “low level” that can be exactly mapped. To improve the results, we relied on the methodology we considered before, which is by considering the synonyms or entry terms in MeSH®, performing a manual mapping for the terms at the “high level” and then applying a semi-automatic mapping to the terms at the “low level”. The semi-automatic mapping strategy compares a MeSH® term with an ICD term from the “low level” by applying a lexical matching, and if an ICD term can match a MeSH® term, it will be mapped to the same MeSH® term its parent was mapped to (Figure 36). Using this strategy, the results can cover approximately all the ICD terms. Moreover, by considering the synonyms in MeSH®, we were able to get mappings that are not included in Bioportal mappings nor in UMLS mappings. An example of these mappings is the ICD term “Intrauterine synechia” which was mapped to the synonym of the MeSH® term “Gynatresia” by an exact lexical mapping (Figure 37).

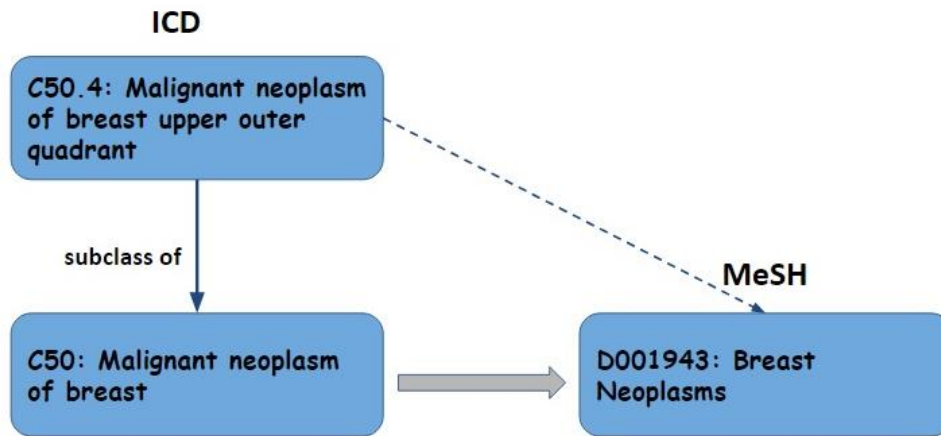


Figure 36. An ICD term is mapped to the MeSH® term that its superclass is mapped to.

<u>Mesh</u>		<u>ICD10</u>
<b>MeSH Heading</b>	Gynatresia	<b>N85.6</b>
<b>Entry Term</b>	Asherman Syndrome	Intrauterine synechia
<b>Entry Term</b>	Asherman's Syndrome	
<b>Entry Term</b>	Intrauterine Synechia	Score=1
<b>Entry Term</b>	Uterine Synechia	
<b>Unique ID</b>	D006175	

Figure 37. An ICD term matches an entry term (synonym) of a MeSH® term.

The results of ICD-MeSH® mappings were saved in tabular format files and they are also available on the web page of the project. Figure 38 shows a glimpse of the ICD terms as well at the “low level” mapped to MeSH® terms.



C91.3	Prolymphocytic leukaemia	D015463	Prolymphocytic Leukemia	Lex
C91.1	Chronic lymphocytic leukaemia	D015451	Chronic Lymphocytic Leukemia	Lex
C91.7	Other lymphoid leukaemia	D007945	Lymphoid Leukemia	Lex
C91.4	Hairy-cell leukaemia	D007943	Hairy Cell Leukemia	Lex
C91.9	Lymphoid leukaemia unspecified	D007945	Lymphoid Leukemia	Lex
D47.2	Monoclonal gammopathy	D010265	Monoclonal Gammopathy	Lex
C63.2	Malignant neoplasm of scrotum	D001943	Genital Neoplasms_Male	Stru
C18.9	Malignant neoplasm of colon unspecified	D001932	Colonic Neoplasms	Stru
C16.9	Malignant neoplasm of stomach unspecified	D001943	Stomach Neoplasms	Stru
C75.9	Malignant neoplasm of endocrine gland unspe	D004701	Endocrine Gland Neoplasms	Stru

Figure 38. A glimpse of the ICD terms at the "low level" mapped to MeSH® terms.

## 5.4. Discussion

The key objective of modern medical research applications is to provide clinicians and researchers access to prior molecular information existing in biological databases in order to promote the investigation of molecular mechanisms of diseases. In clinical care systems, diseases are defined by diagnostic codes using the International Classification of Diseases (ICD) system to study disease patterns. Understanding pathway and molecular information underlying diseases provide insights into disease etiology. This information is distributed across several biomedical databases and resources. Diseases in such resources are sometimes named differently than in clinical systems. Matching disease names defined by the ICD and biomedical resources facilitates the integration of the clinical and biomedical domains.

In general, the interpretation of experimental data typically involves the comparison of clinical and biological data by physicians and biologists with existing datasets and reference knowledge bases [153]. This integration is supported in its logic-based characterization of cancer types, but also in drugs, molecular pathways, and the comparative anatomy of experimental organisms [154]. Mapping disease names in clinical codes to their corresponding names in the NCIT provides a bridge to link clinical data to pathway information in the NCIT. We started to map ICD terms to NCIT terms using a lexical matching approach. Therefore, we developed a lexical mapping module. The approach of this module is based on comparing and matching two words and the characters that constitute the words using string similarity metrics. Each metric solves one problem and calculates a score. To cover the issues that can be solved by each metric, we integrated these metrics to calculate one aggregated score. The score value is in the range between 0 and 1. A score =1 corresponds to an exact matching between words. The lexical mapping can provide exact mappings but only for a small fraction of the ICD terms and mostly the ones that are classified at the highest level/abstract levels. Since our work focuses on cancers and mapping terms more efficiently, we limited the data in the ICD and the NCIT to disease terms that are classified as neoplasms. In the ICD, the neoplasms are classified under the “Neoplasms” chapter at multiple levels. We used the OBA service to extract the terms that are at the two lowest levels. We considered the lowest level as “low level” (descendants) and the higher level as “high level” (ancestors). In addition, we exploited the structure and the properties of the NCIT ontology to limit the terms to the terms that have the “Neoplastic Process” property using the OBA functionality. Hence, we tried to improve the results by using additional approaches. The second approach was to use synonyms since in some cases an ICD term can be exactly mapped to a synonym term instead of a preferred term

in NCIT. The third approach we applied was the structural strategy. The ICD terms of the “low level” are concrete terms which makes it hard to lexically match them with other terms. The overall approach we applied is defined in two steps. In the first step, we map the ICD terms of the “high level” to NCIT preferred and synonym terms, we keep the mappings of the terms that are lexically exactly mapped, and we perform a manual mapping for the remaining terms. The second step considers the mapping of terms of the “low level” using a semi-automatic mapping approach. We start by applying a lexical mapping between the ICD terms and NCIT preferred terms and their synonyms, we keep the mappings that have a score value = 1 which denotes exact matching. For the terms that have a score below 1, we apply a structural mapping by considering the mappings of the ancestors. A child term in the ICD is mapped to the NCIT term that its parent was mapped to.

Moreover, using the rich relations between the entities of different domains in the NCIT provides the ability to create functions that can fulfill our aim of associating possible pathways that affect certain diseases. We used the NCIT ontology structure to develop specific functions that can model disease pathways. We implemented these functions into a plugin in the OBA service that provides access to the NCIT ontology structure. Using these functions, we were able to create queries that can return possible pathways associated with diseases based on genes. We started by extracting gene-disease associations using a relationship between genes and diseases that exists in ontology. We implemented a function that queries the genes associated with a certain disease. Using another relationship between gene and pathways, which also exists in the ontology, we extracted the gene-pathway associations. These associations and the implemented functions allowed us to develop a function to query the possible pathways associated with a certain disease.

Furthermore, using diverse molecular and pathway information resources can offer more biological insights and extensive molecular information since data is curated differently in different databases. The Human Proteome Survey Database (HumanPSD) is a catalog that provides associations of human proteins with diseases, and clinical trials. The diseases in HumanPSD are defined by MeSH® terms. To link ICD terms to MeSH® terms we tried at first to get existing mappings provided by some resources such as Bioportal and UMLS. Each resource uses a different mapping strategy that covers a small fraction of mappings between ICD and MeSH® terms. To provide more possible mappings and try to have our own mappings, we applied our mapping strategies. We used the same approach that we applied to map ICD and NCIT terms. We applied a lexical mapping to the terms of the “high level” of the

ICD terms in the “Neoplasms” chapter. Then we performed a manual mapping to complete the mapping of the terms that didn’t exactly match MeSH® terms and their synonyms. After that, we applied the semi-automatic approach to the ICD terms of the “low level”. Using these strategies, we were able to map most of the ICD terms that cover the “Neoplasms” chapter.

Terminological mapping is a task that is essential to integrate biomedical information between different resources and to enable queries to different biomedical data sources. The challenge of terminological mapping approaches is that no method is optimal. Additionally, applying one mapping strategy is not enough sometimes to get efficient results and to map every term in a particular data source. In some resources, data is represented in a hierarchical structure such as ontologies. These structures are helpful for data integration. Therefore, using the structural strategy is useful to improve mapping results. Human intervention is also required since the machine cannot recognize some terms that have various definitions. Consequently, combining several strategies can provide more efficient results. We made our mapping results available on the web page of our project to be used elsewhere to integrate ICD data with any other biomedical resources, other than the NCIT ontology and the HumanPSD, that include pathway information and use the NCIT and MeSH® terms to define diseases. Moreover, comparing and combining mappings from different resources could also provide more valuable results such as mappings from Bioportal and UMLS. We could not yet benchmark the results because benchmarking needs an intervention of specialists like clinicians, to validate them but we didn’t have the chance to get any intervention.

## 6 Biomedical Word Embedding

### 6.1. Introduction

With the enormous amount of biomedical literature and the rapid growth of the number of new publications, the wealth of scientific knowledge represented in free text is increasing remarkably. There has been much interest in developing techniques that are capable of identifying, extracting, managing, integrating, and utilizing this knowledge from unstructured data and discovering hidden or implicit information. This “hidden” information can be used for predicting function and discovering new biological mechanisms in particular when it is combined with experimental data analysis [69].

The task of identifying biological entity terms like genes, diseases, drugs; is a fundamental step to make use of the information reported in biomedical literature. When bio-entities are recognized, the next step would be the identification of potential relations between them. Uncovering potential relationships can have a gigantic impact on health care decisions. For example, two different genes reported in two different texts but in a similar context, a functional association might suggest between the corresponding genes within the same pathway or the same phenotype.

Biomedical text articles are rich with valuable information that is written in human natural language. Natural language is complex and entails ambiguous semantics that exists at all levels. One challenge in language comprehension is the ability to interpret the meaning of words in a sentence in a way that is consistent with the context.

Text mining is the commonly used process for analyzing and exploring vast amounts of data from unstructured text, in order to convert it into structured data, derive valuable insights, and mine knowledge. A lot of text mining techniques have been used in the field of biomedicine. Several studies have concentrated on using supervised natural language processing to extract knowledge from scientific literature and to improve the use of this knowledge such as GENIES [11], Textpresso [12], and ChemDataExtractor [13], which requires extensive manually labeled datasets for training. Text data is considered to be high-dimensional data. The term “dimensionality” refers to the number of unique terms in a set of documents. The more unique terms in a text document we have, the higher the dimensionality will be.

Deep learning has demonstrated superior performance on a wide variety of text mining tasks. One of the most important techniques of NLP is the representation of words in a text corpus as high-dimensional vectors (word embeddings) by preserving the syntactic and semantic relationships between them [15]. Word Embedding is one of the fundamental applications that transform human language meaningfully into a numerical form so that computers can handle it. Word Embedding is a feature learning technique in natural language processing that reduces the dimensionality of textual data by mapping words into distributed representations in an n-dimensional space. Hence, words with similar meanings will have representations that are close together in the embedding vector space.

Various word embedding models and pre-trained models have been recently published online and applied to several biomedical tasks of NLP [15][16][17]. Wang *et al.* [18] evaluated the performance of word embeddings trained using four different corpora types namely biomedical literature, clinical notes, Wikipedia, and news articles. Smalheiser *et al.* [19] presented a novel unsupervised method to represent words or phrases as low dimensional vectors based on the word co-occurrence frequency and the similarity between words. Most of the word embeddings are usually trained in the word2vec [15] or GloVe [16] model, which uses information about the co-occurrence of each word with its surrounding words, its contexts, to represent it in a distinct vector and disregards the word redundancy and synonymy. *word2Vec* [15] is one of the most popular word representation implementations, that can capture the meaning of words and similarities between words based on the context.

Some studies [155][156] have recently suggested that the integration of text corpora with domain knowledge can be advantageous in improving the word embedding quality. There is ample biomedical knowledge data in the biomedical domain, such as the medical subject headings (MeSH®), which could be examined to supplement the literature's textual data. Using such biomedical domain knowledge would intuitively enhance word embedding quality in such a way to help to capture the semantics of specific concepts.

Moreover, one of the main NLP applications in biomedical research is extracting biomedical entities and their relations. A biological system is made up of multiple interdependent functional components like genes, diseases, medicines, cells, etc. Biological networks are used extensively to explain the interactions between these components. Biological network analysis focuses on network development, network-learning representation and making predictions about biological networks. Deep learning-based approaches have recently been developed to

solve various biomedical problems, such as classification of skin cancer, cell structure, and function modeling, prediction of the transcription factor, and prediction of DNase hypersensitivity. Research has shown that deep learning models can learn arbitrarily complex relationships with existing integrated knowledge from heterogeneous data sets.

In this thesis, we used the word2vec technique to generate biomedical word embeddings in two approaches. The first approach was to generate a biomedical embedding from a preprocessed text corpus to extract disease-drug associations. The second approach was to build and process our corpus from PubMed. I developed a pipeline to process a text of any domain using text cleaning procedures and to generate word2vec embeddings. Using the pipeline, I applied different preprocessing strategies for comparison purposes. In addition, I developed a web service to facilitate the exploration of biomedical concepts. Besides, we have demonstrated that the relationships between entities that have a similar representation in the embedding are biologically meaningful by comparing their relationships to existing knowledge in biomedical databases. I assessed the effect of corpus size on the variability of word representations. We trained Graph-CNN on breast cancer gene expression data with gene networks derived from the generated embeddings in order to predict the occurrence of metastatic events. Our results showed that the gene embedding network is biologically meaningful and performs well when using it as integrated knowledge in machine learning tasks. We then explained predictions of Graph-CNN by deriving subnetworks with relevant genes.

### **6.1.1. Natural Language Processing Techniques and Challenges**

Natural language processing (NLP) goes back to the principles identified in Turing's classic 1950 paper about "Computing machinery and intelligence" [157][69].

Natural language is a human language, as opposed to computer language [158]. It is complex and entails ambiguous semantics that exists at all levels. NLP allows the computer to interact with the human language naturally in order to infer meaning from it. NLP techniques are used on various issues from speech recognition, translation of languages, classification of documents to extraction of information [159].

Semantics usually refers to the sense of language [69]. Semantic ambiguity occurs when a word or phrase has more than one and unrelated meanings (polysemy). This can be a crucial issue in text mining. For example, *TP53* can refer to a tumor protein that is essential for regulating cell division and preventing tumor formation, or to a gene that provides instructions for making this

protein. Or a related phenomenon when different words have similar meanings (synonymy). For the same gene example, *TP53* can also be known as *P53* in biomedical literature.

Another challenge in language comprehension is the ability to interpret the meaning of words in a sentence that is consistent with the context. It is very crucial to take into consideration that the term meaning is highly context-dependent. In a statement, if two terms are reported together, that doesn't always imply a connection between them. For instance, a statement like "The risk of diseases, including diabetes and leukemia, was investigated." [69] does not indicate a functional relationship between diabetes and leukemia.

Computers cannot understand intuitively natural language as humans do [72]. They can deal with structured data like database tables and ontologies, but it is impossible for them to understand what the language is really saying. Sure, computers can tell whether two words are the same or not but dealing with these diverse linguistic phenomena makes it hard for them to distinguish between words or understand their real meanings.

Natural Language Processing (NLP) is a branch of Artificial Intelligence in computer science that is used to enable computers to understand human language, process it, and make sense of it in a valuable manner. It is a set of methods that enables computers to interact with humans, and understand natural languages spoken by them.

Natural Language Processing techniques typically rely on Machine Learning, specifically deep learning, to model human language and resolve the ambiguity. They consist of algorithms to analyze large blocks of text and perform computations to infer relevant information. The Natural Language Processing components include general tasks including sentence segmentation, tokenization, lemmatization, and named entity recognition. Natural languages are generally composed of sentences, and each sentence is a sequence of tokens, where each token represents a word or a punctuation mark. The initial step of natural language processing is identifying words.

NLP addresses the issues of identifying the relevant terms keywords that after validation enter the standardized data and improve it in order to promote clinical decision-making.

Natural Language Processing (NLP) methods, for example, parsers and part-of-speech taggers, have been used to identify biological entities and dynamic interactions between them from text [160].

NLP is being used in many fields, including medicine. The utilization of NLP in the medical field is very significant, as it provides a new level of functionality for health care applications [161]. NLP systems are used for various purposes, including support for decision making,



infectious disease monitoring, automatic encoding, quality control, and patient data indexing [161]. Another important benefit of NLP technology is that it can be used to standardize reports from different organizations and applications since the same automated framework encodes clinical data in heterogeneous reports in a consistent manner that promotes interoperability [161].

## **6.2. Materials and Methods**

### **6.2.1. Word Embedding**

Since machine learning techniques play a major role in the world of text processing and text analysis, it is crucial to make the learning models deal with text data in an advanced computational way. Deep learning has demonstrated superior performance on a wide variety of text mining tasks. Deep learning models are designed to learn from numerical data to perform any sort of job.

In various tasks of NLPs such as information retrieval, the ability to maintain semantic or syntactic similarities between words has been shown to be very helpful [162].

Word Embedding is one of the fundamental applications that transform human language meaningfully into a numerical form so that computers can handle it. Word Embedding is a feature learning technique in natural language processing that reduces the dimensionality of textual data by mapping words into distributed representations in an n-dimensional space. These representations are vectors of real numbers and they are called “Embeddings”. Each word in a vocabulary is represented by a real-valued vector in a predefined vector space. A vocabulary is a set of unique words forming a particular document. In addition to the dimensionality reduction of textual data, it is important to make computers understand the meanings of words and how different words are related to each other. The word embedding technique is able to capture a word context in a text fragment, its semantic relationships, and syntactic similarity to other words that have a similar context. That makes words with similar meanings have similar vector representations in the embedding space.

The benefit of word embedding is the use of dense low-dimensional vectors that are learned in a way that resembles a neural network. A wide variety of applications are available to map words to informative vector representations. Word2vec [163][15], the famous model, learns word representation based on each word’s context that is formed from the surrounding words. Glove [16], uses word-word co-occurrence matrix factorization techniques, and the co-occurrence is also defined upon local context-based learning.

Word embeddings have been applied in several NLP tasks like named entity recognition [164], information extraction (IE) [165][166], machine translation [167], and sentiment analysis [168][169].

## 6.2.2. Word2vec

There has been quite a development over the last couple of decades in constructing word-embedding representations. Word embeddings are distributed word representations where each word is assigned by a real-valued vector.

*Word2vec* [163][15] is one of the most powerful and computationally efficient implementations to create word embeddings, due to its performance and training speed. It was developed by Mikolov at Google in 2013 [15]. It is a shallow neural network that uses two distinct models: ‘CBOW’ and ‘Skip Gram’. It is an unsupervised way to generate word vectors from a raw text corpus by learning syntactic and semantic representations of words. The *word2vec* model uses information about the co-occurrence of each word with its surrounding words to represent it in a distinct vector in a vector space and it detects mathematically their contextual similarity (Figure 39).

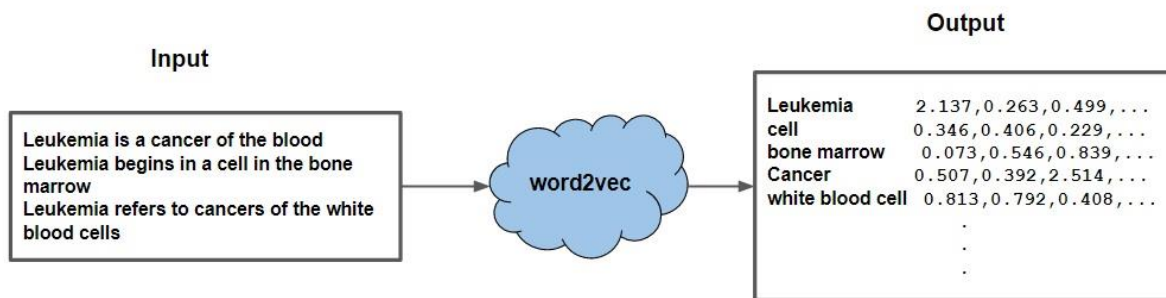


Figure 39. Word2vec converts unique words in a document to distinct real-valued vectors.

Words with similar contexts occupy close spatial positions. The number of context words for a target word is defined by a “window size” which is the maximum distance between a target word and words that surround the target word (default =5) [170]. The similarities between words are estimated using the cosine similarity metric that calculates the cosine of the angle between two words vectors (Figure 40) with the following formula:

$$\text{similarity}(A,B) = \cos(\theta) = \frac{A \times B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Where,

- $A$  and  $B$  are two vectors.
- $A_i$  and  $B_i$  are components of vectors  $A$  and  $B$ .

- The similarity varies between -1 to 1. -1 means least similar, 1 means most similar, and 0 indicates orthogonality.
- In-between values suggest an intermediate similarity or dissimilarity

The cosine distance is defined as:  $1 - \text{similarity}(A,B)$

If two vectors are exactly the same the angle between them is equal to 0, thus the similarity is 1 and the distance is 0.

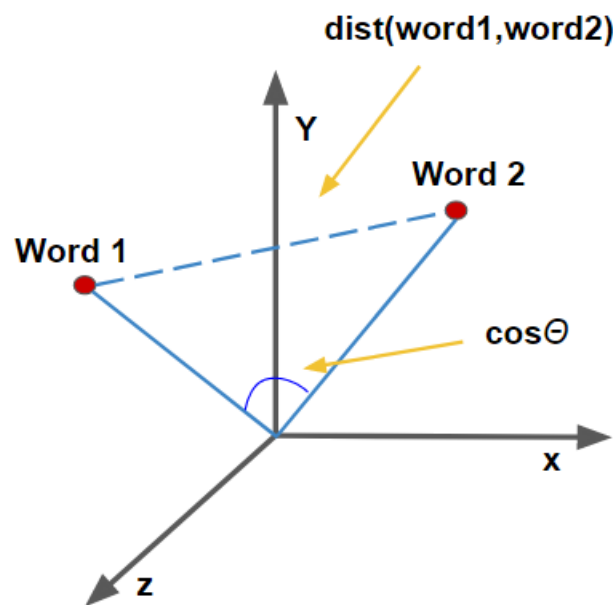


Figure 40. Visual representation of Euclidean distance ( $d$ ) and cosine similarity ( $\theta$ ) between word1 and word2.

The model takes as input a large text and creates a vocabulary  $V$  of unique words in a text. Each word in the vocabulary  $V$  is represented as a one-hot encoded vector (binary vector) according to its position in the vocabulary. For example, if we have the vocabulary  $V$  (word1, word2, word3, word4, word5) “1” is placed in the corresponding position of each word in the vocabulary (Figure 41). The model consists of a two-layer neural net that processes text. The input vector of each word is a hot-encoded vector with dimensions  $1 \times V$ , where  $V$  represents

the number of words in the vocabulary. The hidden layer contains  $N$  neurons and has a dimension that corresponds to the size of the word embedding or the neurons (default = 300). An embedding vector is calculated from the input vector using a matrix of weights for inputs ( $V \times 300$ ) with  $V$  rows and 300 columns (one for every hidden neuron). Then, the output vector is a probability distribution that is calculated from the embedding vector and another matrix of weights for outputs. The output neurons use a Softmax regression classifier to generate an output between 0 and 1 and the sum of all these output values will add up to 1 [171].

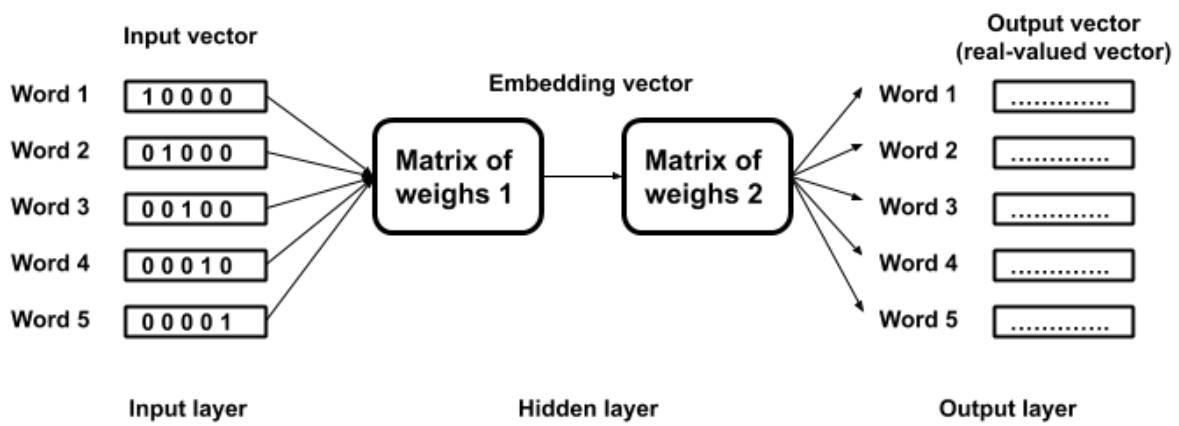


Figure 41. Word2vec Architecture.

The architecture of the two models that word2vec is built with is the same, but the prediction motif for the similarity between words is different.

The continuous Bag-of-Words model (CBOW) learns to predict a target word from source context surrounding words (Figure 42). It calculates a probability vector of a target word from the output vector.

Each element of vocabulary-dimensional vector  $v$  is a probability that a word in the vocabulary becomes a target word. The probability can therefore be calculated by a dot product of the one-hot vector that represents the output vector word and the target center.

Contrary, the Skip-gram model's training aim is to find word representations in a sentence or a document that are useful to predict the surrounding words [15]. It calculates the probability of the vector of a context word. Each element of the vocabulary-dimensional vector is the probability that the word in the vocabulary appears to be a context word in position  $c$ . Thus, the probability can be calculated by a dot product of the one-hot vector that represents a context word at a particular position and the output vector.

CBOW treats the entire context as one observation and it tends to be an efficient approach for smaller datasets. On the contrary, every context-target pair is treated by skip-gram as a new observation, which makes it better suited for larger datasets.

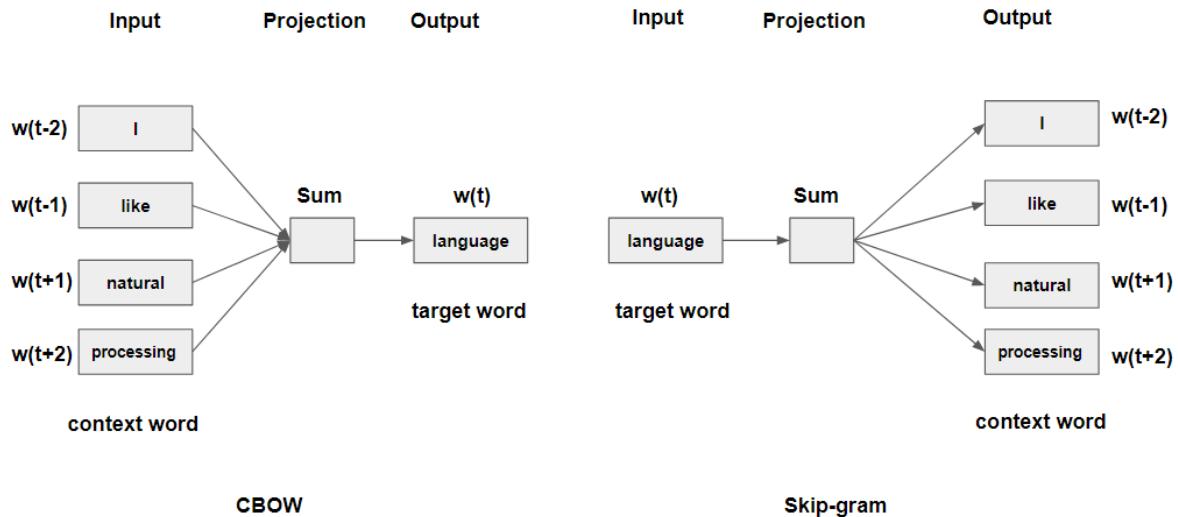


Figure 42. The model architectures of CBOW and Skip-gram [172].

In both CBOW and Skip-Gram, word2vec has two types of algorithms: Hierarchical Softmax and Negative Sampling. Word2vec uses Hierarchical Softmax and Negative Sampling to optimize output layer computation and to accelerate the model training [173].

- **Hierarchical Softmax**

CBOW and skip-gram use the softmax operation to compute the conditional probability for generating the context word or the target word. Softmax is a function that transforms a K float number vector into a distribution of probabilities by first 'squashing' the values to be in a range of [0.0,1.0] [173], to normalize them after that in a way to make the sum equal to 1 (Figure 43). All this is done by Softmax while keeping the relative order of the input float number, thus big input numbers always have a high probability mass for the output distribution [174].

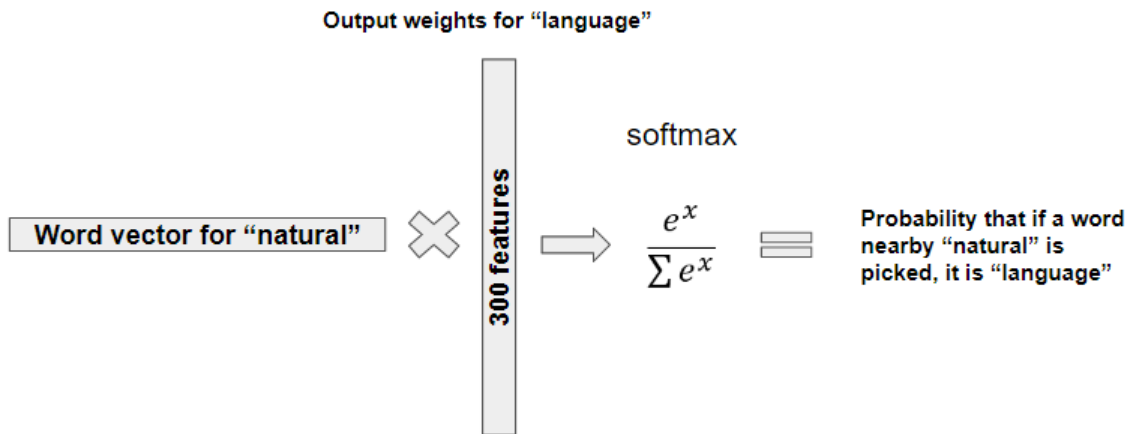


Figure 43. Calculating the probability of the output neuron for a word.

Nevertheless, the training time for the softmax increases linearly with the number of potential outcomes, rendering the approach inappropriate for broad vocabularies [175]. In addition, it is computationally costly for each iteration to measure the softmax likelihood for each word with a vocabulary of a large number of words [176].

In the context of the language models, Hierarchical softmax was proposed by Morin and Bengio [177] to accelerate training, following Goodman's previous work (2001) [178]. The idea is to break down the softmax layer into a binary tree with the vocabulary words at its leaves, such that the probability of a word given a context can be decomposed into probabilities of selecting the right child at each node from the root node to that leaf along the path. This reduces the number of necessary updates from a linear to a logarithmic term in the vocabulary size [175]. By using hierarchical softmax the training complexity is reduced from  $O(V)$  to  $O(\log(V))$  where  $V$  is the number of words in a vocabulary.

- **Negative Sampling**

In Skip-Gram, for each training sample, weight matrices of the neural network are updated to correctly predict the output. The Skip-gram neural network has a large number of weights, which are all slightly modified by each of the trillions of samples.

Assuming there are 10,000 unique words in the vocabulary of a training corpus ( $V = 10,000$ ) and the hidden layer is of 300 dimensions ( $N = 300$ ). Thus, the output weight matrix contains 3,000,000 neurons ( $W_{output}$ ) which can be changed for each training sample. As the corpus size is very large, it is not realistic to update 3 M neurons in terms of computational efficiency for

each training sample. Negative sampling tackles this by updating for each training sample only a small part of the output neurons.

In negative sampling, random  $K$  negative samples are selected using a “unigram distribution”, where more common words are likely to be chosen as negative samples.  $K$  is an empirically tunable hyper-parameter with a typical range of [5,20] [179].  $K$  negative samples are randomly chosen from a  $Pn(w)$  noise distribution, for each training sample with a positive pair: word and positive context  $(w, c_{pos})$ , then in the output matrix of weights ( $W_{output}$ ), the model updates  $(K+1) \times N$  neurons.  $N$  represents the hidden layer ( $h$ ) dimension or the word vector size. +1 is for a positive sample [180].

If  $K=9$  is specified,  $(9+1) \times 300=3000$  neurons are updated by the model, which is only 0.1% of the 3M neurons in  $W_{output}$  according to the above assumption [180]. It is much cheaper in computational terms than the original Skip-Gram, and yet it retains word vectors consistency.

### **6.2.3. Word Embedding Generated using a Preprocessed Text Corpus**

Traditional word2vec methods are by design unsupervised and do not require domain knowledge.

In a recent study, *Ghosh et al.* [181] motivated Dis2Vec, a vocabulary driven word2vec approach to generate word embeddings that are disease-specific from an unstructured text corpus.

Dis2Vec was developed to add information on disease vocabulary as prior knowledge with the aim of generating disease specific word embeddings. They demonstrated the outperformance of the Dis2Vec model by comparing it to other conventional word2vec approaches on tasks for disease characterization. In a study which has been lately conducted in Rzhetsky’s lab [182] at the University of Chicago, they developed a new Named Entity Recognition Ontology (NERO) primarily to describe entities in biomedical text. The ontology considers various ambiguity levels and bridges several scientific sublanguages such as biochemistry, molecular biology, genetics and medicine). A large biomedical corpus was annotated using this ontology in order to facilitate tasks of biomedical natural language processing and machine learning. The Named Entity Recognition Ontology (NERO) and the annotated corpus aim to cover all entity types that might exist in biomedical literature.

By using the implementation of Dis2Vec, we developed a word embedding trained on the corpus they developed that covers sentences from MEDLINE®, Reuters and Wikipedia



referenced articles or abstracts. We integrated a disease vocabulary generated from the entities annotated to the ‘disease’ class in the ontology. We visualized disease and drug embeddings to capture meaningful associations.

#### 6.2.4. Biomedical Embeddings Generated from PubMed/MEDLINE® Abstracts

We used the *word2vec* implementation to generate a biomedical embedding using MEDLINE® abstracts that are linked to the HumanPSD database components. The process of generating embeddings that represent biomedical concepts in a low dimension consists of several steps. We started at first with a pre-processing phase to clean and normalize the text before feeding it into the training model. We generated two embedding versions for comparison purposes. The process of our word embedding is shown in Figure 44. We developed a pipeline to process the text corpus and to generate word2vec embedding. The implementation methods are based on Gensim [183] which is a Python library for unsupervised topic modeling and natural language processing. Moreover, we developed functions to query and explore biomedical concepts and their relations in the resulting embedding. Based on these functions, we developed a web service to facilitate the exploration using an interactive user interface. We describe in more detail the steps of this work in the following sections.

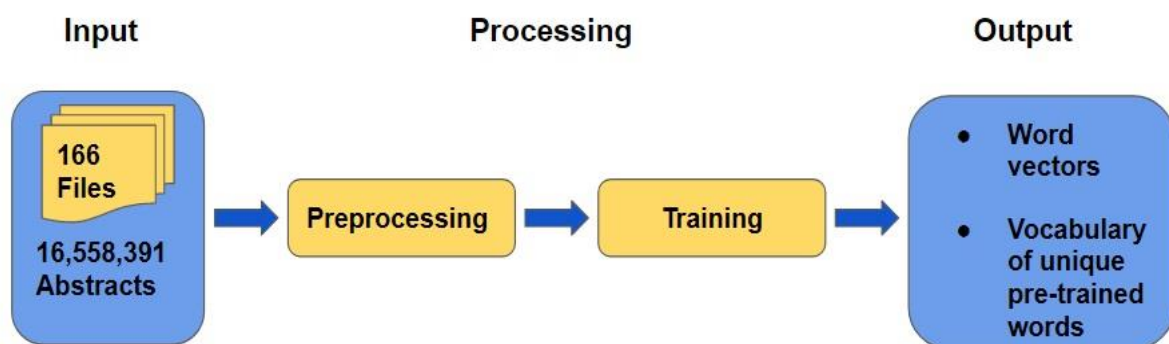


Figure 44. Embedding development workflow. Text processing starts by reading the abstracts as sentences. Preprocessing strategies are applied during a preprocessing phase. The preprocessed corpus is used for training. The output of the training model consists of the word vectors and the vocabulary of pretrained unique words in the corpus.

### 6.2.4.1. Text Preprocessing

Text data needs to be cleaned and normalized before being fed into machine learning models. This process is known as ‘Text Pre-processing’. The pre-processing phase of our work consists of multiple steps. It starts by reading text files from a directory, where a directory is given as input/argument.

We used 166 files containing in total 16,558,093 abstracts. An abstract is a summary that provides readers with a quick and direct overview of an article. It is framed from the key statements of the introduction, methods, results, and discussion sections. So, it should entail the important information that can help the reader to learn about a topic of his interest and to infer relevant information.

Each file consists of several abstracts, one abstract per line. The directory of all the files should be the input of the preprocessing phase. The directory can only contain files with format: .bz2, .gz, and text files. The model in Gensim requires that the input must provide sentences sequentially. This means that there is no need to keep everything in memory: we can provide one sentence, process it, forget it, and load another sentence. Then instead of loading everything into an in-memory list, the input is processed file by file, and line by line. This is called an iterator. Any further preprocessing procedures can be done inside the iterator. Words must be already preprocessed, separated by whitespace, and form a corpus that consists of a bunch of word lists to be fed into the word2vec model.

Each abstract is considered one sentence. A sentence is divided into a list of tokens/words. This process is known as ‘Tokenization’. Tokenization is a crucial part of converting text to numerical data that machine learning models need to be trained and make a prediction. The output of “word tokenization” is a list of words for each sentence, it is a format for better understanding and processing text in machine learning applications. The word lists are then provided as input for further cleaning steps. We used the module ‘word\_tokenize’ from the NLTK library [184] to split the sentences. NLTK stands for Natural Language Toolkit. It is one of the most powerful Python packages that consists of a set of the most common natural language algorithms like part-of-speech tagging, tokenization, named entity recognition, and sentiment analysis.

The ‘word\_tokenize’ module breaks a text fragment by white space and treats punctuation as a separate token as well, which facilitates the removal of punctuation later if desired.

Example (sentence source: <https://www.tocris.com/cell-biology#:~:text=Cell%20biology%20is%20the%20study,to%20understanding%20many%20disease%20states>):

```
• >>> ['Cell biology is the study of the formation, structure, function, communication and death of a cell']
•
• >>> ['Cell', 'biology', 'is', 'the', 'study', 'of', 'the', 'formation', ',', 'structure', ',', 'function', ',', 'communication', 'and', 'death', 'of', 'a', 'cell']
```

After this step, the normalization procedures start, and they are mainly lowercase transformation and lemmatization. Changing words to lowercase prevents considering the same word as two different words during training. Lemmatization reduces a word to its morphological root, which is known as the lemma, by removing inflectional endings and considering the context.

In biomedicine, main terms are often represented by several words like ‘zinc finger protein’.

To create a distributed representation for words that captures their meanings, it is important to identify not just single words but also phrases (multi-word).

It is a fundamental step to generate phrases from sentences. In Gensim [183], the ‘Phrases’ module uses a model that is a simple statistical analysis, where phrases are created based on relative counts (n-grams counts) using the following formula:

$$score(w_i, w_j) = \frac{count(w_i, w_j) - \delta}{count(w_i) \times count(w_j)}$$

Where,

- $count(w_i, w_j)$  is the number of co-occurrences for phrase “ $w_i w_j$ ”
- $count(w_i)$  is the number of occurrences for word  $w_i$
- $count(w_j)$  is the number of occurrences for word  $w_j$
- $\delta$  is used as discounting coefficient which avoids the formation of phrases composed of very rare words.

It is a scoring function that detects words that appear frequently together using some tunable thresholds to decide some token-pairs (Figure 45). The bigrams (two-word phrases) with a score that exceeds a chosen threshold, are used as phrases. Example:

```
sentences = [['lung', 'cancer', 'is', 'a', 'type', 'of', 'cancer', 'that', 'begins', 'in', 'the', 'lungs'],  
             ['lung', 'cancer', 'is', 'one', 'of', 'the', 'most', 'common', 'and', 'serious', 'types', 'of', 'cancer'],  
             ['there', 'are', 'different', 'types', 'of', 'lung', 'cancer']]
```

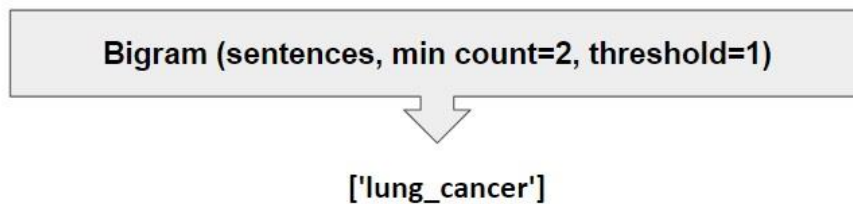


Figure 45. Bigram identification example. “Bigram” is a function in Gensim to create phrases. “sentences” is a list of 3 sentences. “min count” is the minimum value of the total collected count of bigrams. “threshold” is the minimum score for a bigram to be taken into account. “lung cancer” is the identified bigram/phrase which appeared in the 3 sentences.

Subsequently, further cleaning is done by filtering out useless forms and words like stop words, punctuation, and numerical forms (Figure 46). All these filtrations are optional and depend on the purpose of the embedding.

Stop words like “am, a, is, are, this, an, the, etc.” are commonly used words that appear more frequently in text without adding much meaning to a sentence context. Considering them might lead to noisy data and take up memory and time. They can be easily removed by storing a list of these stop words. The NLTK (Natural Language Toolkit) library in python has already a list of stop words stored in different languages.

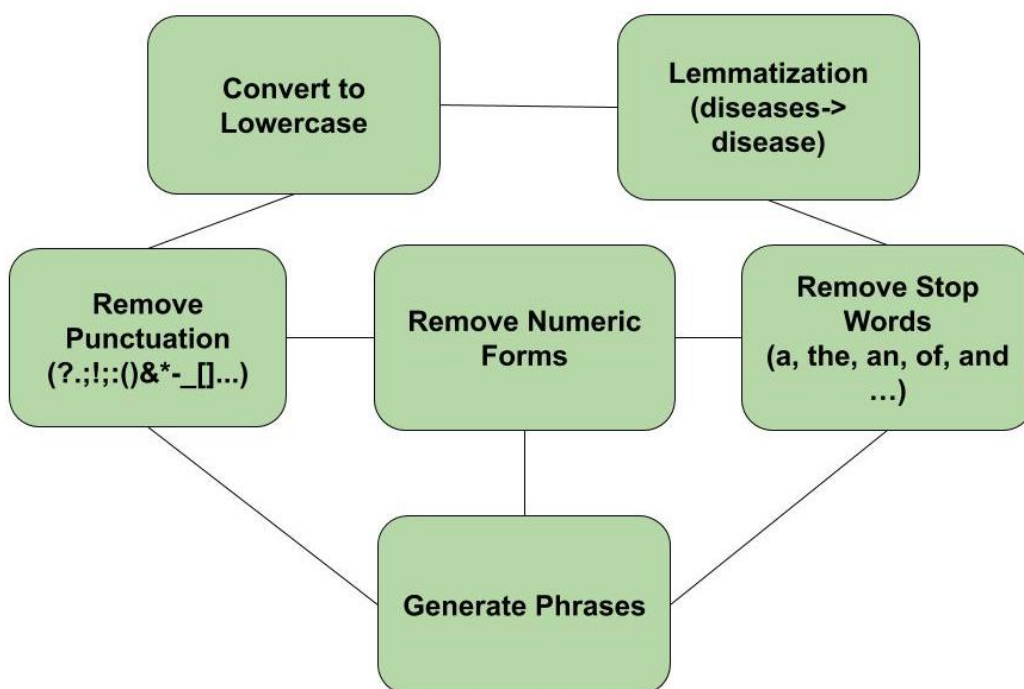


Figure 46. Preprocessing procedures.

Usually, stop words are removed before identifying phrases. In biomedical text, it is a bit challenging when dealing with some stop words since some words can form a part of compound words. For Example, “Vitamin A” is an open compound word where the stop word ‘A’ is an essential stem for the meaning of the word. “Vitamin A” is the name of a group of fat-soluble retinoids that are stored in the liver, including retinol, retinal, and retinyl esters [185][186]. It has its specific roles in maintaining some body functions. Removing ‘A’ leads to a general organic molecule without specifying its type and makes the model classify its context according to the word ‘Vitamin’. In our processing, we considered this procedure as a sensitive case and we dealt with it by changing the conventional chronological preprocessing steps. In order to keep the meaning of the composition of such words, we identified them by generating phrases. Another case we tackled which is also limited to the intended purpose, is the synonymy. Since one of our aims is to get the biological entities that are similar to other entities in order to derive biological meanings from the relationships between them based on the context, and to uncover hidden relationships, we substituted the synonyms of biological entities of multiple types (gene, disease, drugs, and pathways) by their preferred (main) terms using external resources. This procedure was done during the preprocessing phase before training to narrow the similarity between the words that share a similar context and are not synonyms. For genes, we used HUGO [187], for diseases we used MeSH® [137] preferred terms, and for drugs, we used

DrugBank [188][189] terms. This was done for one of the versions. Hence, we generated an embedding version that covers synonyms of biomedical concepts (Embedding\_v1), and another version in which synonymous terms of biological entities were excluded (Embedding\_v2).

### **6.2.5. Web Service Development**

We used the same implemented methods introduced in previous sections to generate word embeddings based on a larger corpus consisting of 17,719,608 PubMed abstracts. The number of unique words in the output vocabulary is 3,221,627. We annotated biomedical entity terms in the output vocabulary to filter similarities for the entity type. We used HUGO [187], MeSH® [137] and DrugBank for genes, diseases, and drugs respectively. In addition, we also used the vocabularies provided by the Comparative Toxicogenomics Database (CTD) [190] to annotate more entities from the output vocabulary namely genes, diseases, and chemicals.

To provide access to the embedding information and to facilitate the exploration of biomedical entities, we developed the embedding of biomedical concepts (eBioMeCon) web service. In this section, I present the materials used for building the back end of the web application. The access to embedding information was supplied with functions implemented in Python. The functionality includes searches for the nearest neighbors of an input word, their distances in the vector space, or extraction of the vectors that represent the words, which can be filtered for a concept type of interest (disease, drug, gene, or pathway). The functions are presented in section 6.3.5.

#### **6.2.5.1. REST API**

REST is for Representational State Transfer, and API stands for Application Programming Interface. The REST API approach is used in web services development. An API is a set of rules that allows programs to communicate with one another. The API is created by a software developer creates on a server to enable various clients including browser applications to talk to it. REST is an architectural style that defines a set of rules that determine how the API looks like.

The REST API approach uses the basic HTTP (Hypertext Transfer Protocol) methods to request data. The common methods of HTTP are GET, POST, PUT, and DELETE. HTTP is a protocol that enables documents to be transmitted back and forth on the web. Such protocol

includes rules that specify the messages that should be transmitted, and are proper responses to others [191].

**GET:** retrieves search information results. This request type is the most common type. We can use it to obtain the data we are looking for, which is ready to share the API [192].

**POST:** collects new data to add it to the server. For instance, a new item may be added to your list using this request type [192].

**PUT:** changes current data. For instance, the color of an existing item may be modified. by using this type of request [192].

**DELETE:** deletes existing information [192].

URL (Uniform Resource Locator) is the endpoint where a service can be accessed by a client application to get a piece of data. A URL is called a request and the retrieved data is called a response [193]. A response is conditional to API architecture, technology, and other aspects. It can be in HTML, XML, JSON, or another format. In Python, there are various frameworks or libraries for creating REST API such as CherryPy [194], webpy [195], Bottle [196], Flask [197] and Django [198]. We used the Flask web framework written in Python to build our web API which enables users to fetch data from a server.

#### **6.2.5.2. PubTator Central (PTC)**

PubTator Central (PTC) is a web service that provides automated annotations of biomedical concepts in PubMed abstracts and full text articles of PubMed Central (PMC) [199][200]. The annotations are based on text-mining systems and they are of several types for genes, diseases, genetic variants, chemicals, cell lines, and species[199]. The Web interface of PTC allows users to create complete sets of text in any document and to display annotations. The web interface design supports full-text annotation and features semantic search. The annotations are provided in different formats and can be downloaded via the web interface and a RESTful web service. PTC comprises millions of abstracts and full-text articles. With new articles being published every day, PTC synchronizes with PubMed and PMC-TM. To automatically annotate biomedical concepts presented in the PubMed abstracts, we used the RESTful API provided in the web application. We were able to annotate diseases, genes, chemicals, species and cell lines in 16,493,738 PubMed abstracts.

### **6.2.5.3. Cytoscape.js**

Cytoscape [201] is an open-source tool that is widely used for visualizing molecular interaction networks and analyzing network biology. Cytoscape.js is a JavaScript-based featured graph library. It is highly optimized and compatible with all modern browsers. It can be used in JavaScript environments to analyze and render interactive graphs in a web browser. It has been used to visualize molecular interaction networks. It can be easily integrated into an application. The library includes many valuable features in graph theory. To provide interactive access to the web service, we visualized the results of some functions as networks.

### **6.2.6. Validation of Word Embeddings**

To provide an expedient tool for biological research, relative locations of terms within the vector space of the embedding should exhibit agreement with existing biological knowledge. Our validation addressed protein-protein interactions, signaling pathways and biological processes, drug targets and human diseases, which have been and continue to be of interest in many biomedical research projects. The conducted validation experiments, therefore, examined whether vectors of members within groups defined by respective biological databases featured increased cosine similarities compared to randomly sampled entities. This section and the following subsections are mainly based on our paper [202].

#### **6.2.6.1. Signaling pathways, biological processes and human diseases**

Reactome 72 [203] and TRANSPATH® 2020.2 [147][39] pathway-gene assignments as well as Gene Ontology (GO, release 2020-03-25) [2] biological process-gene assignments were extracted from the geneXplain platform [80] version 6.0. Disease terms covered by the embedding were mapped to 139 groups of the Human Disease Ontology version 2019-05-13 [204] with more than 5 and less than 1000 member diseases. We calculated medians, lower and upper quartiles of cosine similarities for gene pairs within pathways and biological processes with at least 10 and not more than 3000 genes as well as for disease pairs within the 139 disease groups. In addition, we calculated medians, lower and upper quartiles for 2000 randomly sampled gene pairs and for 700 randomly sampled disease pairs that were not contained in selected groups.



### **6.2.6.2. Protein-protein interactions**

Known protein-protein interactions were extracted from Reactome 63 for 4254 genes (16727 interactions) with vector presentations in the embedding. For the purpose of comparison, we sampled 10000 random gene pairs and the same number of gene pairs with known interactions.

### **6.2.6.3. Drug-gene associations**

The DrugBank [188] database combines detailed drug information with comprehensive drug target information. We extracted genes associated with each drug reported in DrugBank with type target. By considering 5234 drugs and their target genes, we created drug pairs based on the common genes that two drugs share in each pair. Drug-gene associations were obtained from DrugBank release 4.5.0 and cosine similarities of 50000 drug pairs with at least one shared target gene were compared to 50000 drug pairs without common target genes. Moreover, to examine the variability of the similarity distribution of drug pairs based on the number of genes they share, we visualized the distribution of three drug pair groups (group1: no genes, group2:  $\leq 5$  genes, group3:  $\leq 9$  genes). In addition, we examined the drug-drug similarities by comparing drugs sharing common pathways of Reactome and TRANSPATH® databases based on genes. Each pathway in both databases was mapped to one or more genes reported that they play a role in such pathways.

## **6.2.7. Examination of Biomedical Embedding Utility**

To further demonstrate the utility of the word-embedding derived networks, I used the same PPI network employed in [209][216] and created other text-mining-based networks. This section and the following subsections are mainly based on our paper [202].

### **6.2.7.1. Breast Cancer Data**

Graph-CNNs were trained on a breast cancer data set compiled by Bayerlová et al. [205]. The data consisted of 10 microarray data sets measured on Affymetrix Human Genome HG-U133 Plus 2.0 and HG-U133A arrays which are provided online by the Gene Expression Omnibus (GEO) [206] under the following accession numbers GSE25066, GSE20685, GSE19615, GSE17907, GSE16446, GSE17705, GSE2603, GSE11121, GSE7390, and GSE6532. The algorithm of the RMA probe summary [207] was applied to normalize each data set separately after which they were combined and further normalized using quantile normalization applied

on all datasets. If more than one probe was associated with a gene, the probe with the highest average value of expression was chosen, leading to 12179 genes on which the GCNN was trained. Training set classes consisted of 576 patients with no metastasis between 5 and 10 years after biopsy and 393 patients with metastasis developed during the first 5 years (similarly in [216]).

### **6.2.7.2. Graph Convolutional Neural Network and Multilayer Perceptron**

The graph CNN [208] captures a graph signal's localized patterns via convolution and pooling operations performed on a graph. The convolution operation is formulated based on the theory of spectral graph utilizing the convolution theorem and graph Fourier transform. The graph convolutional filter can be approximated by a parameterized expansion of Chebyshev polynomials of graph frequencies [208]. Such a filter of polynomial degree localizes the signal pattern in K-hop neighboring nodes. For the pooling operation, the graph is coarsened exploiting a graph clustering technique. *Chereda et al.* [209] applied the graph CNN with the following hyperparameters for learning. Two convolutional layers were used with 32 convolutional filters and polynomial degree 8 per layer. Maximum pooling of size 2 applies to both of the convolutional layers. Two fully connected layers have 512 and 128 units consequently. ReLU (rectified linear unit) activation function was used, and cross-entropy loss was minimized. The application of usual CNN is not straightforward for gene expression data since it is not spatially ordered. Therefore, we applied deep Multi-layer Perceptron implemented in Keras [210], on the same set of genes but without prior knowledge structuring the data. The hyperparameters of our deep neural network are the following: 4 hidden layers and each of them consists of 1024 units with ELU (exponential linear unit) activation function. Cross entropy loss was minimized.

### **6.2.7.3. Study Approach**

The approach of *Chereda et al.* [209] is to structure gene expression data by applying it to prior knowledge on molecular interactions and to feed this structured data as input for the graph CNN deep learning method (Figure 47). The endpoint is to predict the occurrence of a metastatic event for a patient and classify him into metastatic or non-metastatic. The first group corresponds to patients with metastasis developed during the first 5 years and the second concerns patients who are metastasis-free within the first 5 years.

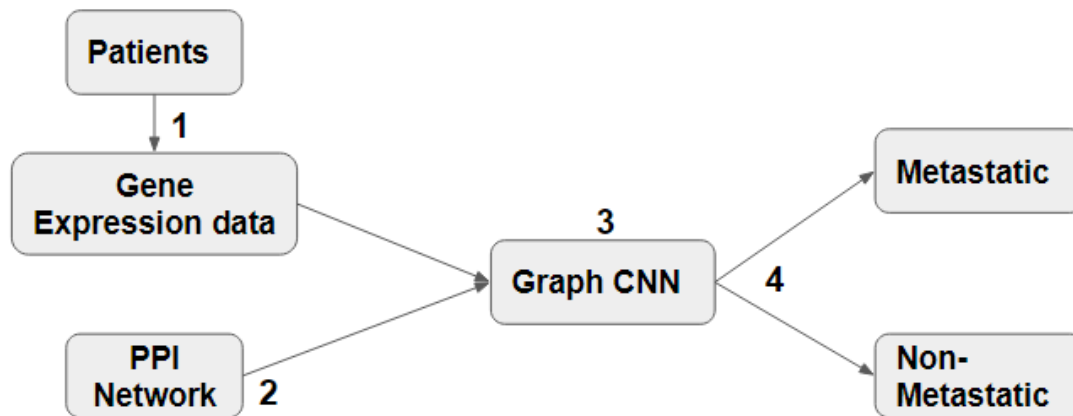


Figure 47. Approach workflow: 1. Patients’ microarray data is preprocessed. 2. Genes are mapped to the vertices of the PPI network. 3. Graph CNN processes gene expression data as graph signals. 4. Graph CNN predicts whether the patient is getting metastases during the first 5 years or not. The figure is based on the approach workflow of *Chereda et al.* represented in [209].

#### 6.2.7.4. PPI Networks

A broad range of machine learning models has been developed to analyze high-throughput datasets with the aim of predicting gene interaction and identifying prognostic biological processes. Recently, biomedical research has shown the ability of deep learning models in learning arbitrarily complex relationships from heterogeneous data sets with existing integrated biological knowledge. This biological knowledge is often represented by interaction networks. The high data dimensionality and the complexity of biological interaction networks are significant analytical challenges for modeling the underlying systems biology. In this section, we present the PPI networks derived from different sources and used as prior knowledge to structure gene expression data.

##### 6.2.7.4.1. Human Protein Reference Database

*Chereda et al.* [209] employed the Human Protein Reference Database (HPRD) protein-protein interaction (PPI) network [211] to structure gene expression data of breast cancer patients. Genes from gene-expression data were mapped to the vertices of the PPI network yielding an undirected graph with 7168 matched vertices consisting of 207 connected components. The main connected component had 6888 vertices, whereas the other 206 components each contained 1 to 4 vertices (similarly in [216]). Since the approach of utilizing prior network

information in Graph CNNs required a connected graph [211] training was carried out on the gene set of the main connected component

#### **6.2.7.4.2. STRING-derived Network**

The STRING database [212] is a collection of protein-protein associations which can be derived from one or more sources such as gene neighborhoods, gene co-occurrence, co-expression, experiments, databases, text-mining, and whose confidence is expressed by an aggregated score computed from scores of the individual interaction sources. We considered the text-mining score as well as the combined score to build weighted protein-protein interaction networks. This way, the classification performance of Graph CNNs trained on the STRING text-mining network could be compared to Graph CNNs with prior knowledge from word2vec embedding-based networks. Likewise, the HPRD PPI, we mapped the genes to the two constructed STRING networks and supplied their main components to the training process. Score thresholds were chosen to obtain a comparable number of vertices as in the HPRD PPI.

#### **6.2.7.4.3. Word2vec-embedding-based Networks**

We created two gene-gene networks (Embedding\_net\_v1 and Embedding\_net\_v2) from the embedding version that excluded synonyms (Embedding\_net\_v1) and another (Embedding\_net\_v2) where word synonyms were considered. Both networks consisted of gene pairs with edges weighted by their cosine similarity values. The cosine similarity threshold was set to 0.65 yielding the Embedding\_net\_v2 network with 10729 genes in 4399 components with the main component covering 6106 vertices and the Embedding\_net\_v1 network with 10730 genes in 4397 connected components with the main component of 6092 vertices. The main connected components of Embedding\_net\_v1 and Embedding\_net\_v2 networks shared 5750 genes, therefore the majority of vertices overlapped.

#### **6.2.7.4.4. BERT-embedding-derived Network**

BERT (Bidirectional Encoder Representations from Transformers) [213] is a model that has been recently developed for contextualized word representations. The main technical innovation of Bert is the use of bidirectional transformers. BERT was pre-trained in English Wikipedia and Books Corpus as a general language representation model. BioBERT [214] is a language representation model based on BERT and designed for biomedical text mining tasks.

It was initialized with the BERT model provided by *Devlin et al.* in 2019 [213] and pre-trained on PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC). We used the pre-trained BioBERT weights of ‘BioBERT-Base v1.0’ that was trained using the same vocabulary of BERT base (12-layer, 768-hidden, 12-heads, 110M parameters) on English text and 200k PubMed abstracts in addition. We converted the pre-trained TensorFlow checkpoint model to Pytorch [215], extracted the numerical vectors of 768 dimensions each, and calculated the cosine similarities between entities to eventually extract a gene-gene network.

#### **6.2.7.5. Graph-CNN trained with PPI Networks**

One of the approaches for validation of the embedding networks is to analyze how the underlying molecular network influences the performance of the machine learning method utilizing prior knowledge. In recent studies [209][216] the Graph-CNN method was applied to the breast cancer dataset introduced in section 6.2.7.1. In order to retain gene expression values non-negative, we subtracted the minimum data value (5.84847) for each cell in the gene expression matrix. If GE data was initially in [5.84847, 14.2014] now it is in [0.0, 8.3529]. The classification accuracy of Graph-CNNs was compared for different sources of network prior information. We compared the influence of several prior knowledge networks on performance: HPRD, Embedding\_net\_v1, Embedding\_net\_v2, String, and BioBERT-based networks. As for embedding networks, we utilized weighted and unweighted (taking into account only the topology) versions. The vertices were mapped to the genes of gene expression data and weighted edges were filtered according to a threshold value. We considered thresholds higher than 0.5 for cosine similarity between vertices. The main connected component of the underlying graph was used to structure the data. The performance was assessed by 10-fold cross-validation. For each of the data splits the model was trained on 9-folds and the classification was evaluated using the 10th fold as a validation set. For each underlying molecular network, the architecture and hyperparameters of Graph-CNN remained the same. For the majority of the cases, Graph-CNN was trained with 100 epochs, but for some versions of prior knowledge, a smaller number of epochs showed better results since the convergence of gradient descent was happening faster. The most common evaluation metrics were used: AUC (Area Under the Curve) ROC (Receiver Operating Characteristics) curve, accuracy, and F1-weighted score. We averaged the metrics over folds and evaluated the means' standard errors.

### 6.2.7.6. Explaining decisions of Graph Convolutional Neural Networks

In order to generalize a machine learning model well, it is important to ensure that its decisions are validated by relevant patterns in input data. Deep learning approaches, like CNNs, are frequently criticized for being “black-box”. Deep Neural network decisions are notoriously hard to interpret. Layer-wise Relevance Propagation (LRP) is a common method used to explain deep neural network predictions. LRP aims at providing an explanation of an output of the neural network in its input domain. LRP establishes back-propagation of the output through the network up until the input layer using the weights of the network and activations produced by the backward.

*Cherera et al.* [216] presented a new method, Graph Layer-wise Relevance Propagation (GLRP), to explain Graph-CNN’s decisions. The method was applied on the breast cancer gene expression data introduced in section 6.7.1, structured by HPRD PPI. GLRP provided patient-specific molecular subnetworks on the PPI basis. We applied the same approach to explain the results of Graph-CNNs trained on the gene expression data structured by the embedding-derived network `Embedding_net_v1` and to assess how an underlying network affects the explanations.

The architecture of Graph-CNN consists of 2 convolutional layers. Two convolutional layers were used with 32 convolutional filters. Maximum pooling of size 2 applies to both of the convolutional layers. Two fully connected layers have 512 and 128 nodes consequently.

Graph-CNN was trained on 90% of data (872 patients) and 10% was saved as a test set (97 patients). The predicted patient data were classified into two groups: metastatic and non-metastatic. The probability of each class is shown by an output neuron of the neural network. We selected from the test set subnetworks of 4 breast cancer patients with a correctly predicted class: 2 metastatic and 2 non-metastatic. GLRP was used to propagate the relevance from the output node of Graph-CNN corresponding to the label that was predicted correctly. For each patient, the relevance for each gene out of 6092 genes in the embedding network was calculated according to the following relevance propagation rule:

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

Where,

- $j$  and  $k$  are two neurons in each successive layer
- $a$  is for the respective neuron activation
- $w$  denotes for the weight between the two neurons

The genes were ranked by relevance value and the most relevant genes were selected from the embedding network to create the subnetworks. Two variants of patient-specific subnetworks were generated by the selection of the embedding network vertices with the highest relevance values: 140 and 200 top-relevant vertices.

## 6.3. Results

In this section, I present the results of the generated word embeddings and the computational analysis performed. The results are divided into two parts, one for the results we produced using a preprocessed corpus and mainly based on our paper [217] (section 6.3.1), and the other part is for the results of the embeddings generated using a corpus that we created and processed. The validation and evaluation results in sections 6.3.3, 6.3.6, 6.3.7, and 6.3.8 are mainly based on our paper [202]. The methods of both parts are based on Gensim.

### 6.3.1. Disease-drug Associations

The result of the generated embedding using the Dis2Vec implementation presented in section 6.2.3, is a vector space representation that positions words that tend to occur in similar contexts of other words more closely to each other. The vocabulary of the resulting embedding encompasses 5,656,455 unique words. Each word has a distinct vector of 300 dimensions. The embedding covers all types of biomedical concepts. The similarities between words are estimated using the cosine similarity metric. These similarities can uncover valuable relationships between biomedical entities from different perspectives such as synonymy relationships, relationships between entities that belong to the same system in case of diseases, or between entities that share the same gene family in case of genes. For example, for the disease entity “eczema” the most similar words are “dermatitis” (similarity value= 0.863) and “atopic dermatitis” (similarity value= 0.858) which are exact synonyms of “eczema”. Moreover, other types of relationships can be revealed by visually detecting entities. By representing the embedding entities into a visualized three-dimensional space, we were able to capture visible disease-drug associations. To capture validated disease-drug associations, we used the 5-min consult data [218] that includes information about disease names that are related to free-text drug information. We processed the free-text information and extracted drugs related to diseases. We mapped the disease and drug names to corresponding words in the embedding by using our lexical developed mapping module introduced in section 5.2.1. We were able to match 174 diseases and 242 drugs. We annotated the diseases and drugs by systems, to be able to capture associations that are within the same systems. For diseases, we used a list of diseases annotated in Rzhetsky’s lab. For drugs, we used the Anatomical Therapeutic Chemical (ATC) classification system [219] to assign systems. The ATC system is a system for classifying drugs based on their active ingredients depending on the system or organ they act on as well as their chemical pharmacological and therapeutic properties [220]. By graphically representing high-dimensional embeddings, we can better visualize, understand



the embedding layers, and highlight words that are nearby in the embedding space. We added the vectors of the annotated diseases and drugs into one array. The labels and the classes were assigned as metadata. The visualization was implemented using Maya [221], a 3D graphics tool, which is used to create assets for interactive visual effects. We visualized the vectors in 3D using principal component analysis (PCA) which is used as a dimensionality reduction technique. This visual representation shown in Figure 48 allowed us to identify some examples of drug-disease associations. Figure 48 displays the representation of diseases (prisms) and drugs (spheres) in different colors that correspond to systems. Figure 49 and Figure 50 illustrate diseases and related drugs in the ‘neoplastic process’ and ‘central nervous’ systems. Figure 51 shows an example of ‘Zollinger-Ellison syndrome’ with related drugs. Figure 53 shows an example of the ‘Amphotericin B’ drug and related diseases. The related diseases and drugs were identified in the 5-min clinical consult. The entities that are visualized and recognized in the 5-min clinical consult are highlighted in Figure 52 and Figure 54. These results are part of a paper entitled “NERO: a biomedical Named Entity (Recognition) Ontology with Large Annotated Corpus Reveals Meaningful Associations Through Text Embedding”, which is under review and it is available as a preprint under the following link:

<https://www.biorxiv.org/content/10.1101/2020.11.05.368969v1.full>.

To evaluate the generated word embeddings based on NERO (Named Entity Recognition Ontology), the newly developed ontology introduced in this chapter (see section 6.2.3), the projections of diseases and related drugs were then compared into the embedding dimensions for severity and gender as disease properties, and for toxicity and expense as drug properties. The properties are not explicitly present in text; however, they are relevant for diagnosis and treatment. The embedded meanings were compared with ground truth data about diseases and drugs. The arithmetic mean of word vectors that represents antonyms in a dimension was taken to construct meaningful dimensions which were used to diagnose their meanings [217]. For more details about the method and the results of the word embeddings’ evaluation based on NERO, please check our preprint paper [217].

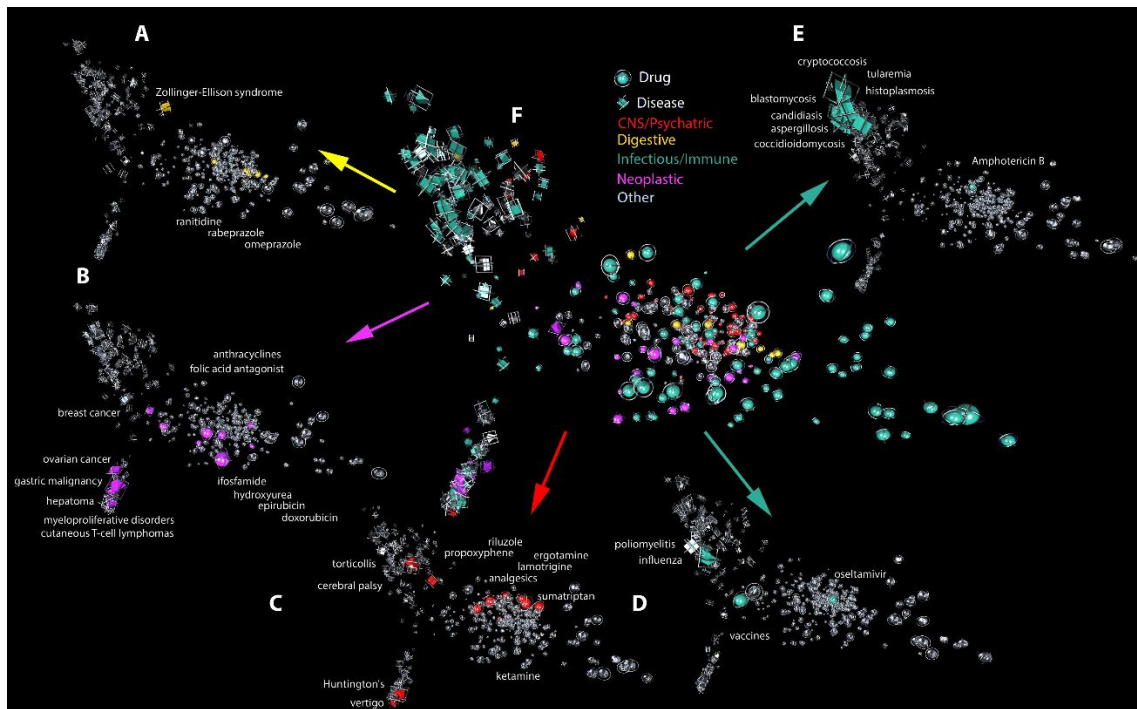


Figure 48. The figure depicts a projection of text embedding into three-dimensional space, with prisms and spheres correspond to named entities referring to diseases and drugs, respectively. The distance between entities is calculated based on the similarity between two-word vectors of 300 dimensions. (Figure from [217]).

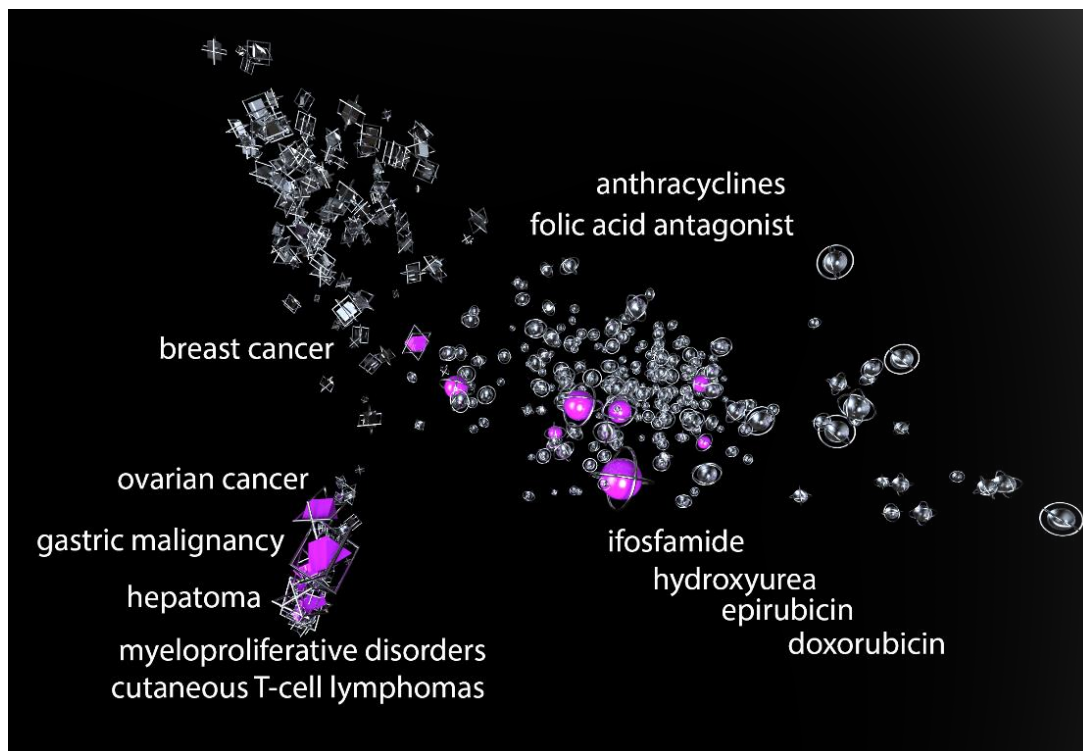


Figure 49. Projection of diseases and drugs embeddings of the 'neoplastic process' system. Prisms and spheres correspond to to diseases and drugs, respectively.

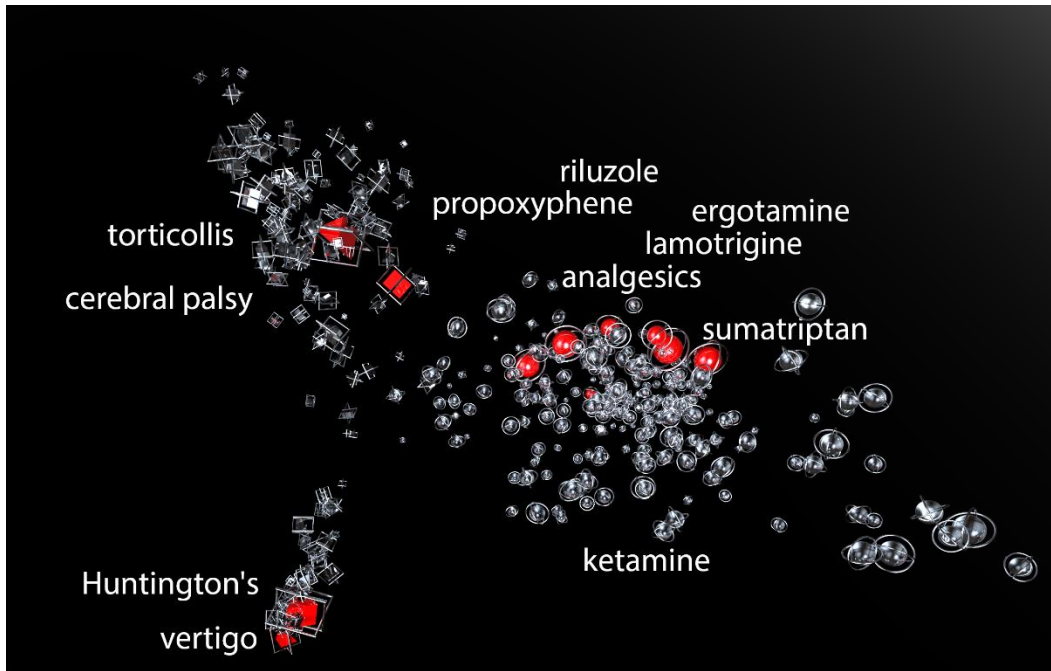


Figure 50. Projection of diseases and drugs of the 'central nervous' system. Prisms and spheres correspond to to diseases and drugs, respectively.

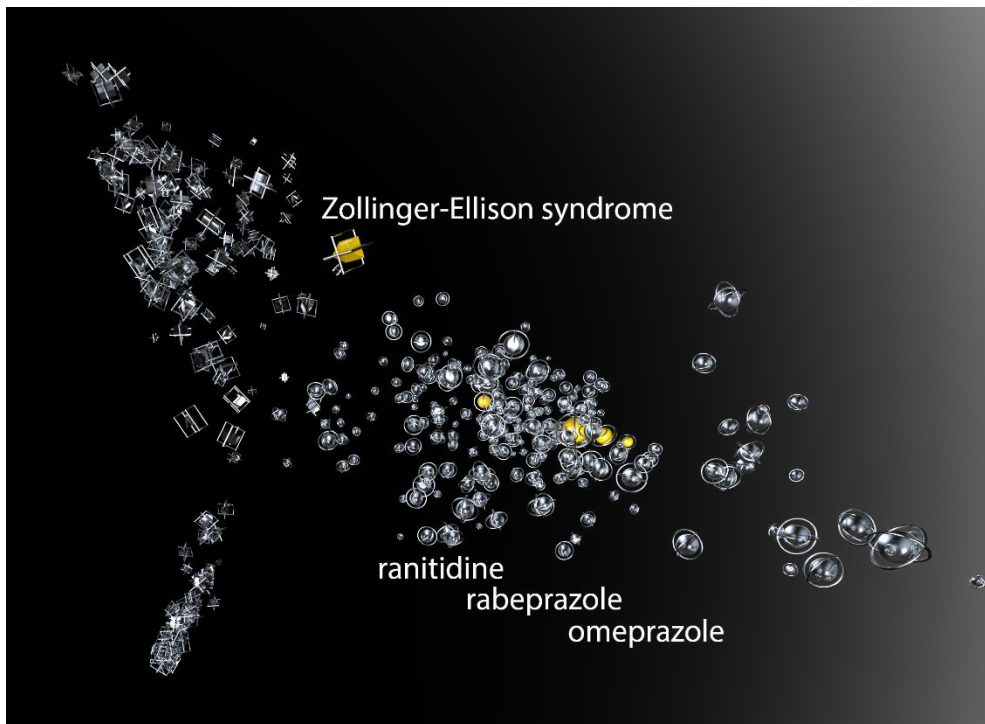


Figure 51. 'Zollinger-Ellison syndrome' and related drugs. The 3 drugs shown in this projection (spheres) are the same drugs identified as drugs related to 'Zollinger-Ellison syndrome' (prism) in the 5-min clinical consult.

doxorubicin	ZOLLINGER-ELLISON SYNDROME
ranitidine	ZOLLINGER-ELLISON SYNDROME
lansoprazole	ZOLLINGER-ELLISON SYNDROME
Doxorubicin	ZOLLINGER-ELLISON SYNDROME
octreotide	ZOLLINGER-ELLISON SYNDROME
Octreotide	ZOLLINGER-ELLISON SYNDROME
rabeprazole	ZOLLINGER-ELLISON SYNDROME
Drugs	ZOLLINGER-ELLISON SYNDROME
PPIs	ZOLLINGER-ELLISON SYNDROME
ampicillin	ZOLLINGER-ELLISON SYNDROME
omeprazole	ZOLLINGER-ELLISON SYNDROME

Figure 52. Drugs related to ‘Zollinger-Ellison syndrome’ in the 5-min clinical consult. The highlighted drug terms are the same terms identified in the embedding projection of Zollinger-Ellison syndrome and their similar drugs illustrated in Figure 48.

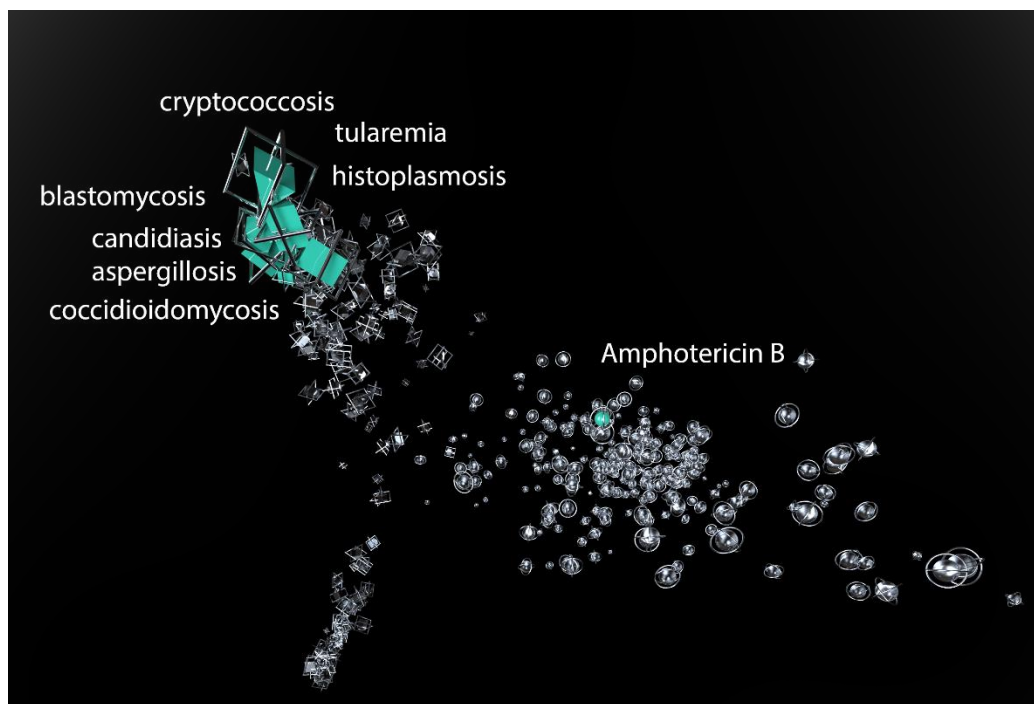


Figure 53. ‘Amphotericin B’ and related diseases. The diseases shown in this projection (prisms) are the same diseases identified as diseases related to ‘Amphotericin B’ (sphere) in the 5-min clinical consult.

Amphotericin B	ARTHRITIS_INFECTIONOUS_GRANULOMATOUS
Amphotericin B	ASPERGILLOSIS
Amphotericin B	BLASTOMYCOSIS
Amphotericin B	CANDIDIASIS
Amphotericin B	CANDIDIASIS_MUCOCUTANEOUS
Amphotericin B	COCCIDIOIDOMYCOSIS
Amphotericin B	CORNEAL_ULCERATION
Amphotericin B	CRYPTOCOCCOSIS
Amphotericin B	HISTOPLASMOSIS
Amphotericin B	LEISHMANIASIS
Amphotericin B	ONYCHOMYCOSIS
Amphotericin B	OTITIS_EXTERNA
Amphotericin B	SPOROTRICHOSIS
Amphotericin B	SUPERFICIAL_THROMBOPHLEBITIS
Amphotericin B	TULAREMIA

Figure 54. Diseases related to ‘Amphotericin B’ in the 5-min clinical consult. The highlighted disease terms are the same terms identified in the embedding projection of ‘Amphotericin B’ and their similar diseases illustrated in Figure 50.

Uncovering valuable biological relationships between entities extracted from text embedding can be helpful to populate an empty ontology structure by contents. Moreover, relationships extracted from the embedding could add more relationships that were not captured in manual annotations.

### 6.3.2. Generated Word Embeddings

Figure 55 shows the results of the nearest neighbors of the “wnt4” gene and “breast\_cancer” in the two embedding versions. The results on the left are from the embedding that covers synonyms (Embedding\_v1). While the ones on the right are from the embedding where the synonyms were substituted (Embedding\_v2). It is observable that the first nearest neighbor of “wnt4” in Embedding\_v1 is “wnt-4” which is an exact synonym, while in Embedding\_v2, it is another gene “wnt7a” that has a biological association with “wnt4” (Figure 55). Similarly, for “breast cancer”, which was substituted by “breast neoplasms”, the first nearest neighbor was also an exact synonym in Embedding\_v1, however, it is another neoplasm “ovarian\_neoplasms” in Embedding\_v2 (Figure 55). These similarities between biological entities can uncover hidden relationships that might not have been yet reported in existing databases such as gene-gene interactions, disease comorbidities, and more types of relationships.

Embedding_v1	Embedding_v2
<p><b>('wnt4')</b>  <b>Out[207]:</b>  [('wnt-4', 0.8259146213531494),  ('rspo1', 0.807041347026825),  ('wnt7a', 0.7983325719833374),  ('fgf9', 0.7910366058349609),  ('wnt9b', 0.7887377738952637),  ('wnt7b', 0.778262734413147),  ('wnt6', 0.7742607593536377),  ('wnt11', 0.7695648074150085),  ('wnt3', 0.763670027256012),  ('wnt2', 0.7498937845230103)]</p>	<p><b>('wnt4')</b>  <b>Out[208]:</b>  [('wnt7a', 0.8113290667533875),  ('wnt7b', 0.7973092794418335),  ('wnt11', 0.7863411903381348),  ('rspo1', 0.7833156585693359),  ('wnt9b', 0.7795987129211426),  ('wnt6', 0.7752448916435242),  ('fgf9', 0.7537680864334106),  ('bmp4', 0.7455331087112427),  ('wnt3', 0.743033766746521),  ('lgr4', 0.7397333383560181)]</p>
<p><b>('breast_cancer')</b>  <b>Out[209]:</b>  [('breast_cancer', 0.8587049841880798),  ('ovarian_cancer', 0.857623815536499),  ('endometrial_cancer', 0.8327959775924683),  ('prostate_cancer', 0.8098835945129395),  ('colorectal_cancer', 0.786318302154541),  ('cervical_cancer', 0.7731184959411621),  ('cancer', 0.7590190172195435),  ('tnbc', 0.7569955587387085),  ('colon_cancer', 0.7559269070625305),  ('ovarian_carcinoma', 0.7554397583007812)]</p>	<p><b>('breast_neoplasms')</b>  <b>Out[212]:</b>  [('ovarian_neoplasms', 0.8639780282974243),  ('endometrial_neoplasms', 0.8336026072502136),  ('prostatic_neoplasms', 0.8241965174674988),  ('colorectal_neoplasms', 0.8044326305389404),  ('colonic_neoplasms', 0.7809585928916931),  ('uterine_cervical_neoplasms', 0.7764290571212769),  ('tnbc', 0.7734637260437012),  ('lung_neoplasms', 0.7650272250175476),  ('ovarian_carcinoma', 0.7490032911300659),  ('pancreatic_neoplasms', 0.74430251121521)]</p>

Figure 55. The first 10 nearest neighbors of 'wnt4' and 'breast neoplasms' obtained using the resulting Embedding\_v1 and Embedding\_v2. The similarity between terms is computed using the cosine similarity metric. The similar terms are ranked by the cosine similarity value from highest to lowest.

In order to be able to access the embedding information and to get more insights into biological relationships, we annotated entities of multiple types from the generated vocabulary using the main terms and their synonyms from the same databases we used to substitute synonyms, of which 17,128 genes, 2,628 diseases, 3,380 drugs, 43 pathways, respectively, are currently covered.

In this part, we present the results of the biomedical embedding we generated. Table 1 shows the numbers of words before and after processing and the number of words in the vocabulary after training. Each word in the vocabulary has a real-valued vector of 300 dimensions. It is remarkable that the number of words after processing is reduced due to the applied preprocessing procedures that remove useless words and merge words into pairs to form phrases where each phrase is considered one word.

Table 1. Embedding results.

<b>Number of words before processing</b>	<b>2,947,203,961</b>
<b>Number of words after processing</b>	<b>1,478,317,457</b>
<b>Words in vocab (unique words)</b>	<b>2,468,093</b>

Figure 56 shows the most frequent words in the embedding based on the number of their occurrences in the biomedical text. It is obvious that ‘gene’ and ‘neoplasms’ are two of the most frequent words in biomedical literature which is not surprising since the main focus of current biomedical research is to report findings related to neoplasms.

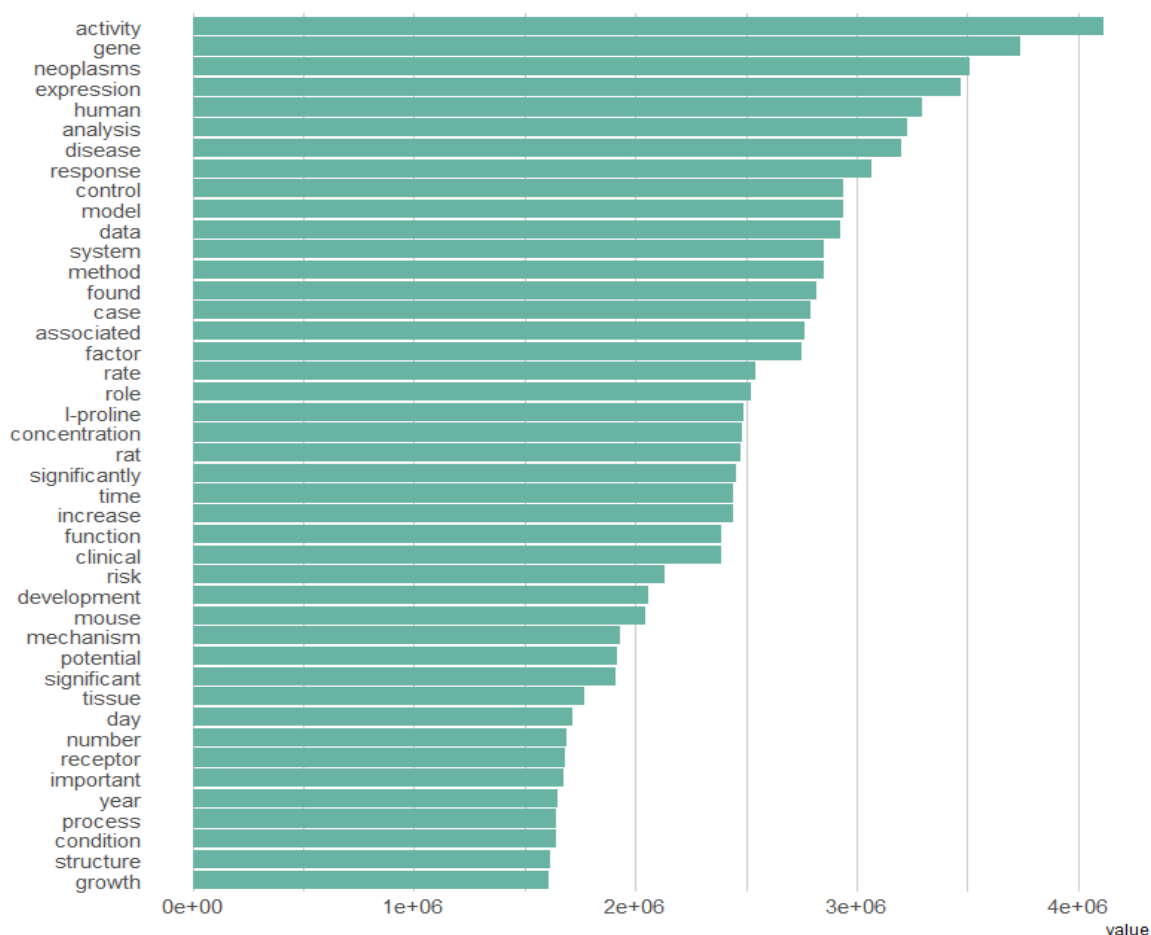


Figure 56. Each word in the output vocabulary has a count property of its frequency in the trained corpus. The frequency of a word is the number of times that a certain word appears in text. In this figure only the most frequent words are displayed. They are ordered from most-frequent to least-frequent.

### 6.3.3. Computational Pipeline for Biomedical Embeddings

For the purpose of pre-processing the text corpus, we implemented a pipeline that conducted the steps depicted in Figure 57. Firstly, in the pre-processing phase, traditional strategies such as lowercasing, lemmatization, and removal of punctuation and numerical forms are applied. Additionally, the “Phrases” function in Gensim is used to detect phrases. We assumed replacing synonymous terms with their main terms can affect the similarity between words in a way to better capture functional relationships between biomedical entities. Thus, for the training phase, we added an optional step in which a dictionary file of terms and their synonyms can be provided in order to update the corpus with substituted terms before using it for the training. In our work, we employed this pipeline to generate two representations of word2vec embeddings for analysis and comparison purposes. For both representations, the same preprocessing strategies were applied to generate the text corpus used for training. However, for one of the representations, we substituted synonymous terms of genes, diseases, drugs, and pathways by their preferred terms in biomedical databases. For training, we used the word2vec implementation in the Gensim [183] Python library with context window size 5, minimum count 5, and 300 neurons which is also the number of generated vector dimensions.

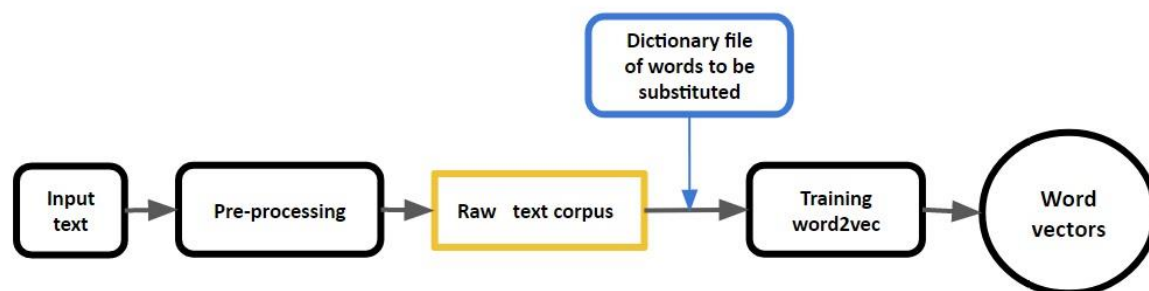


Figure 57. Workflow of the developed computational pipeline.

#### ▪ Preprocessing

The preprocessing phase of this pipeline consists of multiple steps. It starts by reading text files from a directory, where the directory is given as input/argument. Each text will be divided into



sentences by blank lines, where each line is considered one sentence. After converting the text into a corpus of word lists, cleaning procedures are applied, such as lowercase transformation, useless forms removing (stop words, punctuation, and numerical forms), and lemmatization, etc. Subsequently, the identification of phrases starts based on a tunable threshold and the number of token-pairs occurrence in the text that the system uses to identify phrases. For example, if this number =3, the system will identify word pairs that appear together 3 times or more in text. All these filtration procedures are optional and depend on the purpose of the embedding.

In this pipeline, the user has the option to remove stop words during the preprocessing or the training phase. The output of this processing phase is one text file that contains the preprocessed corpus. Figure 58 shows the preprocessing steps as a workflow.

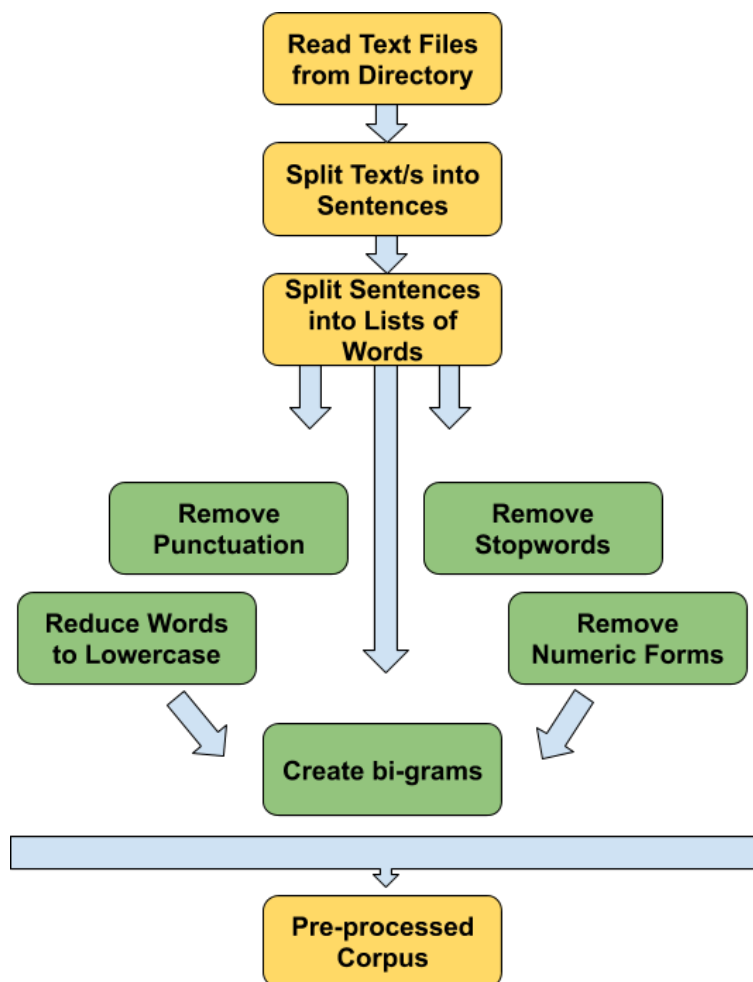


Figure 58. Preprocessing workflow.

- **Training**

Training the *word2vec* model is straightforward. Once the corpus is preprocessed, it can be passed to the model as a sequence of sentences. Before training, a vocabulary of unique words in the corpus is automatically created. The training is applied to the words of the vocabulary by considering the model arguments that can be assigned by the user, otherwise, the arguments are assigned by default values. Word2vec accepts several parameters that affect both training speed and quality. The parameters of the model are tunable such as the “window size” of a word context (default=5), the number of vector dimensions which is the number of neurons that the model uses in the hidden layer (default=300), and the “minimum count” parameter which is the number of times a word needs to appear in the text to be considered in the vocabulary (default=5). The output is the pre-trained model which can be loaded later. Figure 59 shows the training phase steps as a workflow.

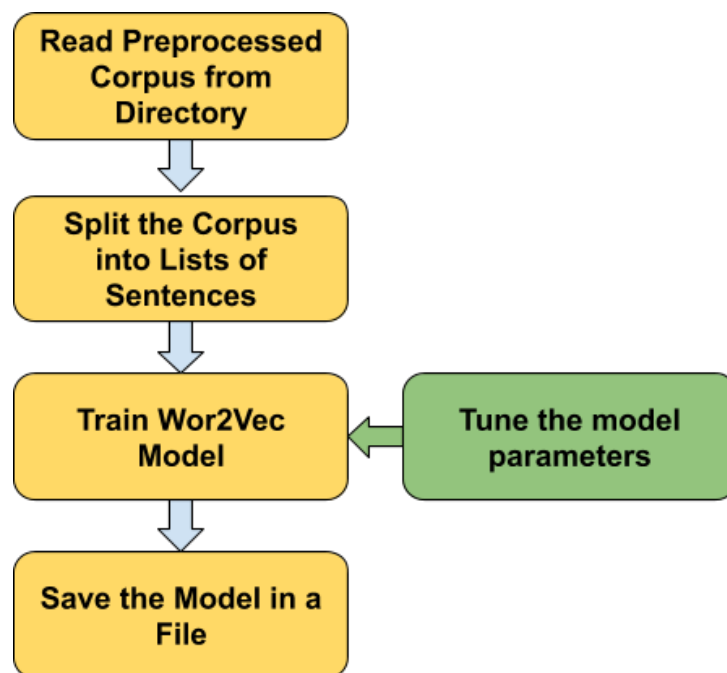


Figure 59. Training workflow.

#### **6.3.4. Assessment of our generated Biomedical Word Embedding**

Embedding is the projection of data into a distributed representation in a space. Visualizing high-dimensional data can help to understand the embedding results and assessing the association of related words in the vector-space. We used TensorBoard Embedding Projector

[222] from TensorFlow [222] to interactively visualize the high-dimensional embeddings. The Embedding Projector provides two widely used data dimensional reduction techniques that facilitate the visualization of complex data: PCA, and t-SNE. PCA is very useful in exploring the internal structure of the embeddings and reveals the most effective dimensions in the data. On the other hand, t-SNE [223] is helpful to discover nearby neighborhoods and to identify clusters, which allows ensuring that the embedding retains the meaning of the data.

In our work, we used the annotated biomedical concepts to select 17128 genes, 2628 diseases, and 3380 drugs, and to visualize the principal components of their embeddings. Figure 60 shows a 3D representation of the embeddings in TensorBoard. Different colors result from metadata (label and class) embeddings. The colors illustrate how clusters are formed. In the visualization, one can click on any point to show the list of its nearest points and their corresponding distances, which illustrates which words the algorithm has learned to be semantically related. It is observable in Figure 60 how the points of the same entity type are grouped closely together. This property provides the ability to identify synonyms or functional relationships. Moreover, in the middle few points from the different types are mixed up. This is useful to capture associations between the different biomedical entities such as disease-gene, disease-drug, gene-drug. Additionally, one can also isolate points that belong to the same class. Figures 61,62,63 show the isolated points of genes, diseases, and drugs with labels, respectively. This isolation permits us to check visually the points of the same class altogether by labels.



Figure 60. The representation of selected genes, diseases and drugs in the embedding space. Each data point represents the learned embedding for a given word. The distance between a data point and its neighbors is the cosine distance. The colors correspond to the 3 classes: gene, diseases and drugs. The red represents genes, the blue is for diseases, and the purple represents drugs.

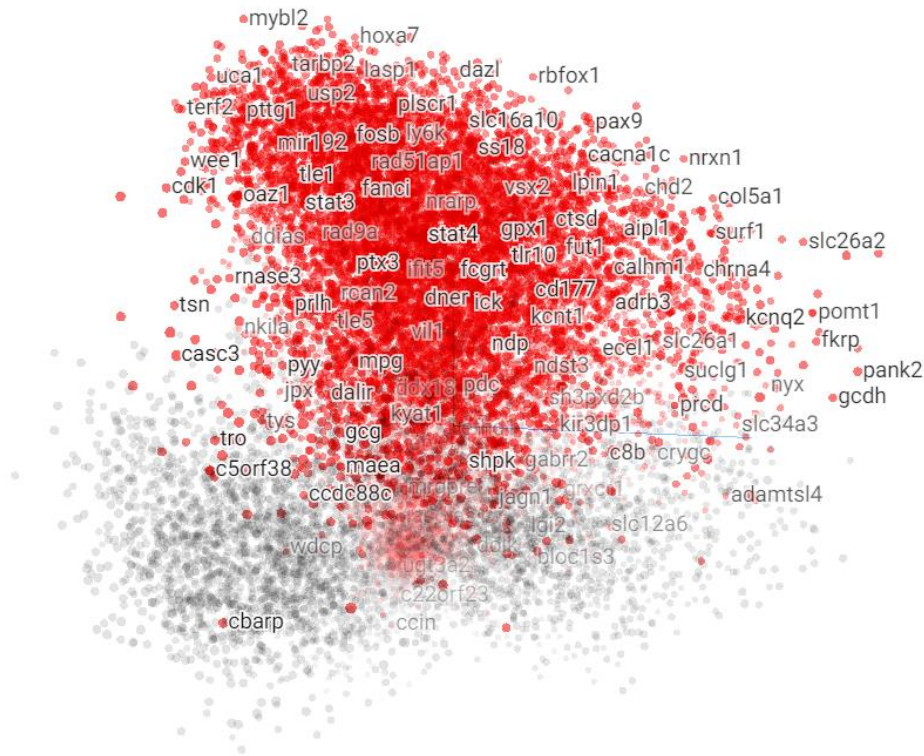


Figure 61. The representation of isolated genes with their respective labels.

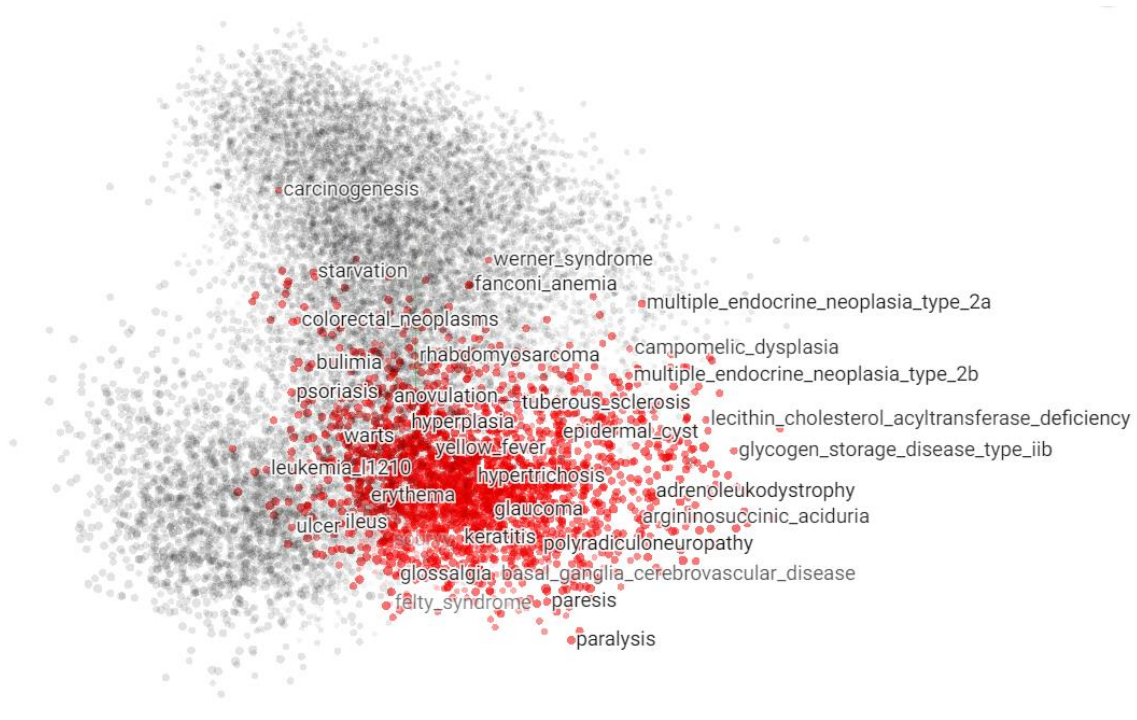


Figure 62. The representation of isolated diseases with their respective labels.

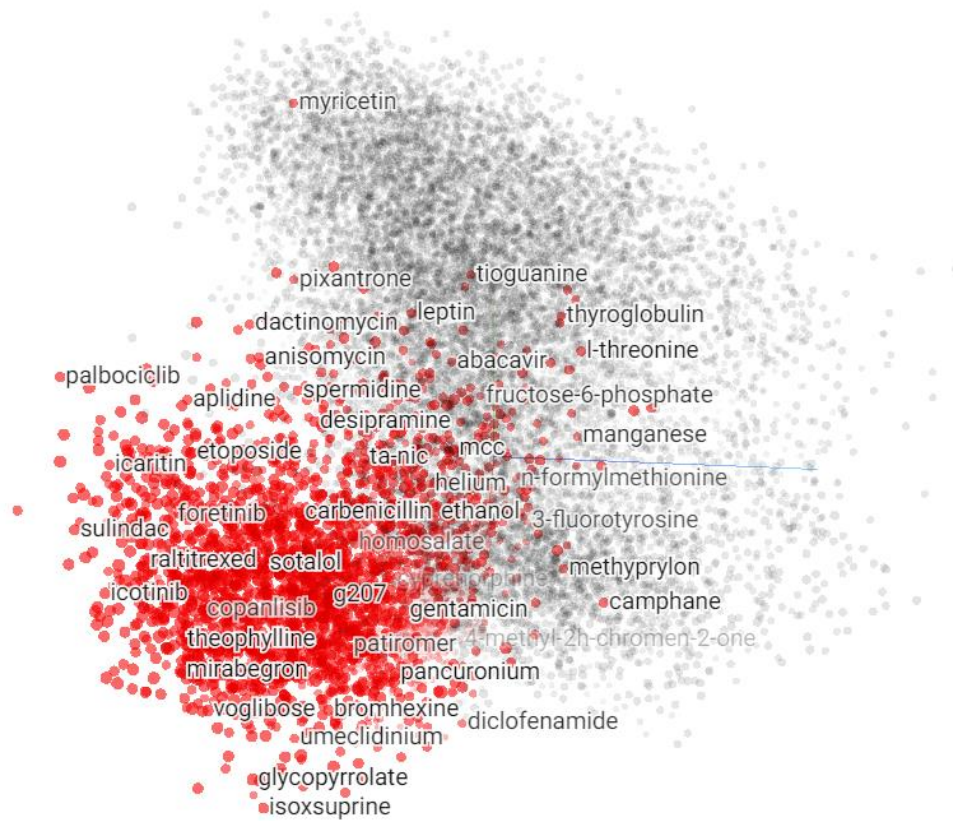


Figure 63. The representation of isolated drugs with their respective labels.

We created a list of 120 randomly selected genes, diseases, and drugs. We employed hierarchical clustering by utilizing the linkage method ‘ward’ in order to examine how good is the Euclidean distance in identifying similar terms. The Ward method tries to minimize the variance within each cluster. It aims to minimize the total variance around cluster centroids. In Figure 64, the dendrogram illustrates how the points are allocated to the clusters. The illustration shows which items belonging to different entity types are clearly grouped together. Similar drugs are grouped in green. Similar genes are in red. Diseases are grouped into one cluster formed by the light blue and the purple sub-clusters.

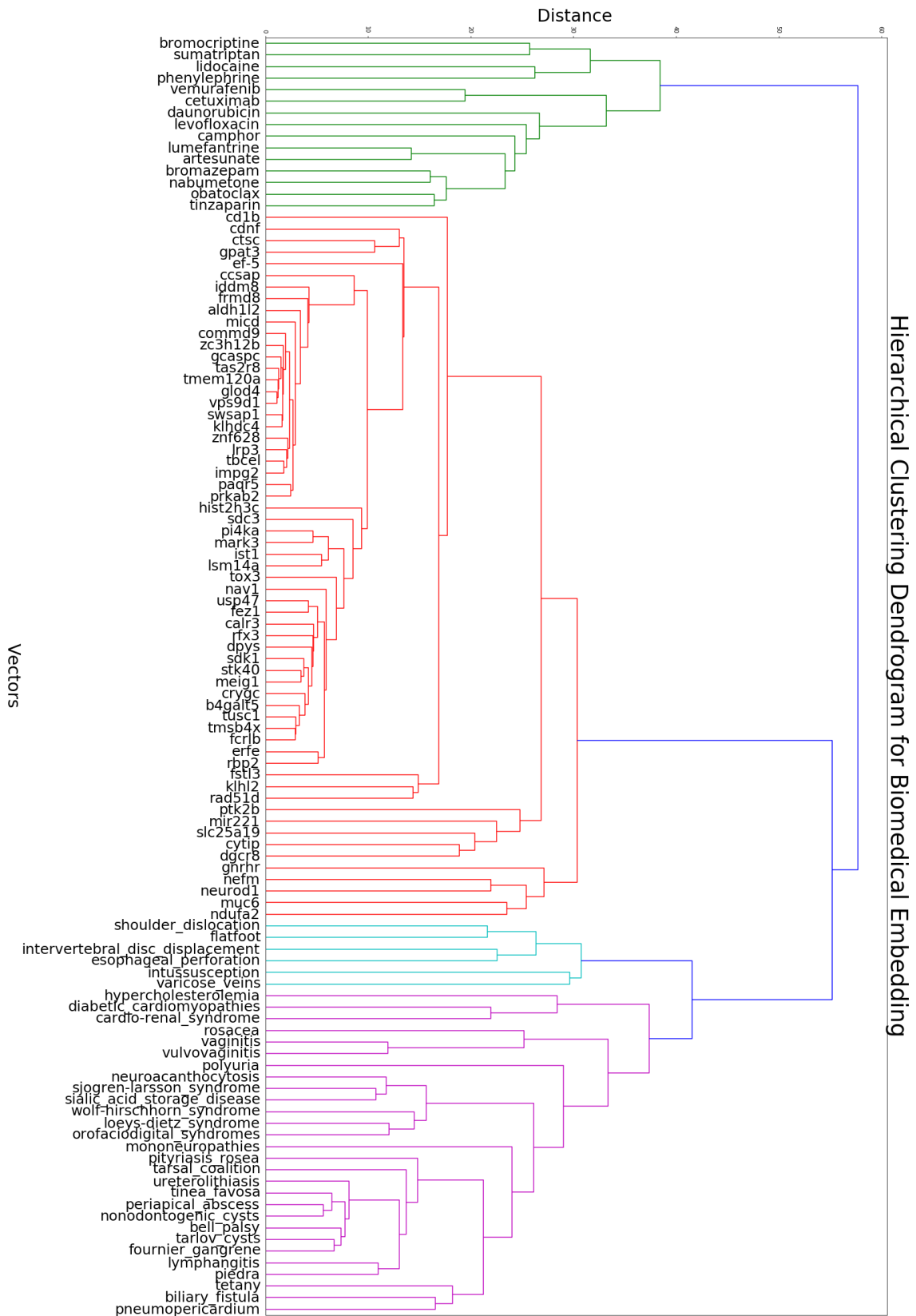


Figure 64. The x-axis consists of the biomedical concepts and y-axis consists of the Euclidean distance between the clusters. Horizontal lines represent merges of clusters. Vertical lines show the clusters formed a new cluster as part of the merge. Horizontal line heights indicate the distance needed to be bridged in order to create a new cluster.

### **6.3.5. eBioMeCon: a web service for querying and exploring biomedical concepts and their relations**

We developed the embedding of biomedical concepts (eBioMeCon) web service that provides access to the embedding information and facilitates the exploration and the querying of biomedical concepts in the embedding in an interactive way. It offers several ways to explore inferred contextual similarities between diseases, drugs, genes, and pathways through graphical and programming interfaces. The data of the web service is based on the embedding version that excludes synonyms. Querying and exploration of this embedding are enhanced by the annotation of entities using biomedical databases and PubTator (see section 6.2.5.2). Our work focused on relations between diseases, drugs, genes, chemicals, cell lines, species, and pathways of which 87,860 genes, 32,314 diseases, 1,198 drugs, 448,539 chemicals, 2,701 cell lines, 32,526 species, 102 pathways respectively, are currently covered by eBioMeCon. The back-end of the web service is implemented in Python and the Flask-RESTful [197] framework for the API (Figure 65). The front-end uses the Bootstrap framework that is based on HTML, JavaScript, and CSS for the presentation and layout (Figure 65). The eBioMeCon provides both a graphical web interface as well as a RESTful API to explore the resulting embedding (Figure 68). The functions of the application can also be accessed programmatically. The functionality includes searches for the nearest neighbors of an input word, their distances in the vector space, or extraction of the vectors that represent the words, which can be filtered for a concept type of interest (disease, drug, gene, chemical, cell line, species or pathway, so that one can focus on relations between, for instance, genes that are related to a particular disease (Figure 66). A user can also create a list of weighted edges from a given list of terms, e.g. gene names, which can be used as a network for further analysis such as prior knowledge in machine learning tasks.



The eBioMeCon is available at <https://ebiomecon.genexplain.com/>.

The source code is available at <https://github.com/genexplain/eBioMeCon>.

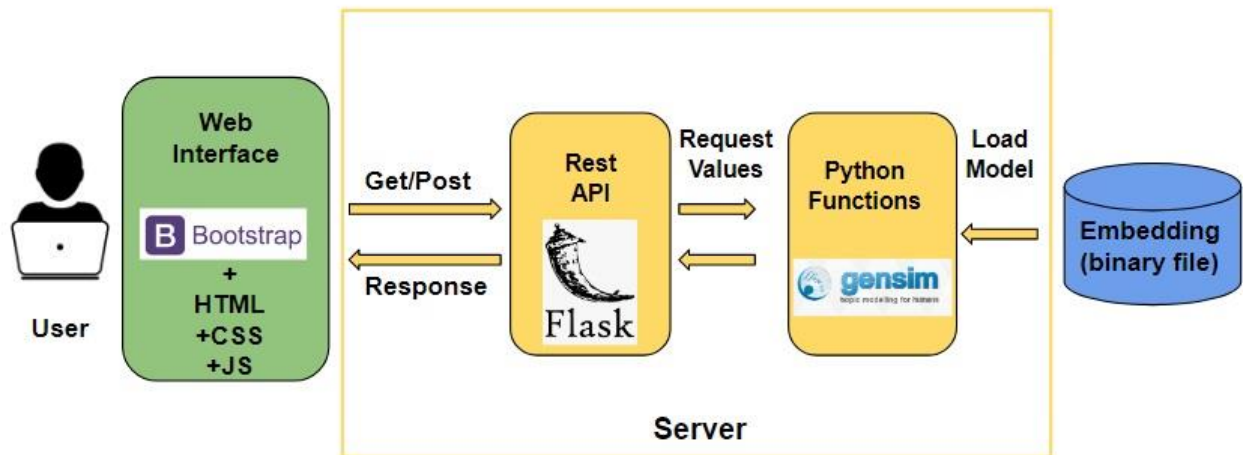


Figure 65. eBioMeCon architecture. The back end of the web service is implemented in Python and the Flask-RESTful framework for the API. The front-end uses the Bootstrap framework that is based on HTML, JavaScript, and CSS for the presentation and layout (<https://github.com/genexplain/eBioMeCon>).

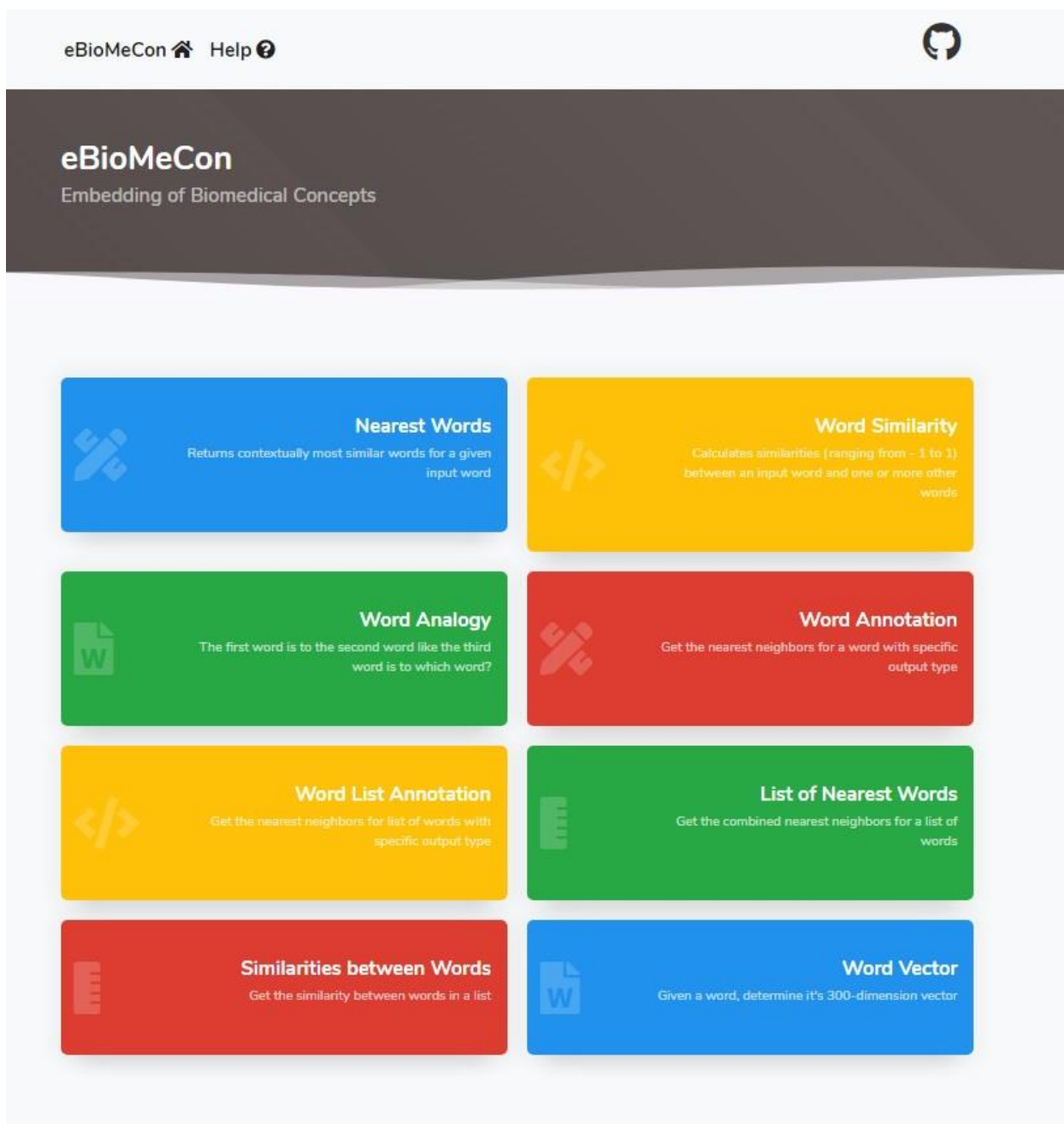


Figure 66. A screenshot of the home page of eBioMeCon. The functions are listed as boxes.

### 6.3.5.1. Web Service Functions

All the functions can be queried using the interactive user interface or URL string queries by providing the corresponding parameters. Both types of queries return responses in HTML. An API query can also be used, and it returns a response in JSON format.

- Nearest Words

This function returns the nearest neighbors that share a similar context in text with a particular word. The result is represented as a network consisting of the input word and its nearest neighbors (Figure 67).

Example string query: [/page\\_nearest?word=mdm2&size=6](/page_nearest?word=mdm2&size=6)

Example API query: [/page\\_nearest/api?word=mdm2&size=6](/page_nearest/api?word=mdm2&size=6)

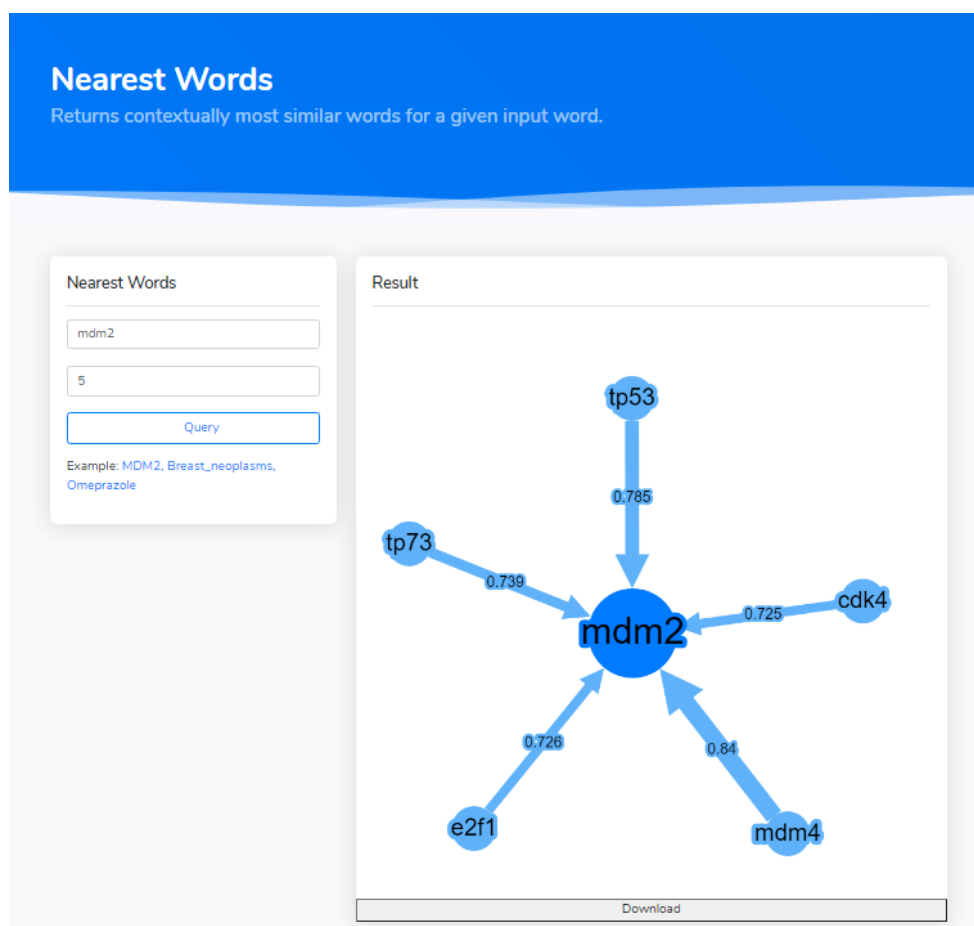


Figure 67. The first 5 nearest neighbors of the 'MDM2' gene. The central node is the input word. The surrounding nodes are its neighbors. The edges are the similarities.

```

{
  - mdm2: [
    - {
      simword: "mdm4",
      value: 0.84
    },
    - {
      simword: "tp53",
      value: 0.785
    },
    - {
      simword: "tp73",
      value: 0.739
    },
    - {
      simword: "e2f1",
      value: 0.726
    },
    - {
      simword: "cdk4",
      value: 0.725
    },
    - {
      simword: "tceal1",
      value: 0.691
    }
  ]
}

```

Figure 68. The nearest neighbors of the 'MDM2' gene with the API response in JSON format.

- Word Similarity

This function calculates the cosine similarity of two given words or a word and a list of words with a similarity value between 1 and -1 (Figure 69). Cosine similarity is a measure that calculates the cosine of the angle between two-word vectors. More detailed explanation about cosine similarity can be found in section 6.2.2.

Example string query: [/page\\_similarity?word=MDM2&simwords=MDM4](/page_similarity?word=MDM2&simwords=MDM4)

Example API query: [/page\\_similarity/api?word=MDM2&simwords=MDM4](/page_similarity/api?word=MDM2&simwords=MDM4)

## Word Similarity

Calculates similarities (ranging from - 1 to 1) between an input word and one or more other words.

### Words Similarity

  
  
  
Example: TP53 --  
MDM2,CHEK2,TP73,CDKN1A

### Result

Word	Target Word	Value
tp53	mdm2	0.785
tp53	chek2	0.638
tp53	tp73	0.765
tp53	cdkn1a	0.667

Figure 69. The similarities between a given word (e.g. TP53 gene) and a list of words (e.g. genes: MDM2,CHEK2,TP73,CDKN1A). The similar words are sorted by similarity value in a descending way.

- Word Analogy

This function checks the semantic analogy between terms. It is used to get a target word "Term 4" that is similar to a particular word "Term 3" according to the semantic similarity between two other words "Term 1" and "Term 2". It performs vector arithmetic: adding the positive vectors (Term 1 and Term 3), subtracting the negative (Term 2), then from that resulting position, listing the known-vectors closest to that angle.  $([Term\ 1 - Term\ 2] + Term\ 3 \sim Term\ 4)$  (Figure 70). Word Analogy can solve analogy questions by calculating the syntactic relationships between word vectors.

Example string

query: [/page\\_analogy?word1=disease&word2=neoplasms&word3=drug&size=5](/page_analogy?word1=disease&word2=neoplasms&word3=drug&size=5)

Example API

query: [/page\\_analogy/api?word1=disease&word2=neoplasms&word3=drug&size=5](/page_analogy/api?word1=disease&word2=neoplasms&word3=drug&size=5)

## Word Analogy

Check the semantic analogy between terms.

((Term 1 – Term 2) + Term 3 ~ Term 4).

A semantic similarity between Term 3 and Term 4, is similar to the one between Term 1 and Term 2.

Word	Value
anticancer_drug	0.634
chemotherapeutic_drug	0.575
anticancer_agent	0.556
antineoplastic_drug	0.539
chemotherapeutic_agent	0.538

Figure 70. Similar terms produced by the word analogy "[disease - neoplasms] + drug =?". The most similar word = "anticancer drug". The similar terms are sorted by similarity value in a descending way.

- Word Annotation and Word List Annotation

These functions work as the "Nearest Words" function, but the user is able to specify the output type of the similar words by choosing one of the biomedical entity types (diseases, drugs, genes, chemicals, cell lines and species) (Figure 71). These functions were developed based on annotating concepts using external biomedical resources.

Example query string: [/word\\_annotate\\_list\\_page?words=MDM2,TP53&type=gene&size=5](#)

Example API query: [/word\\_annotate\\_list\\_page/api?words=MDM2,TP53&type=gene&size=5](#)

## Word Annotation

Get the nearest neighbors for a word with specific output type

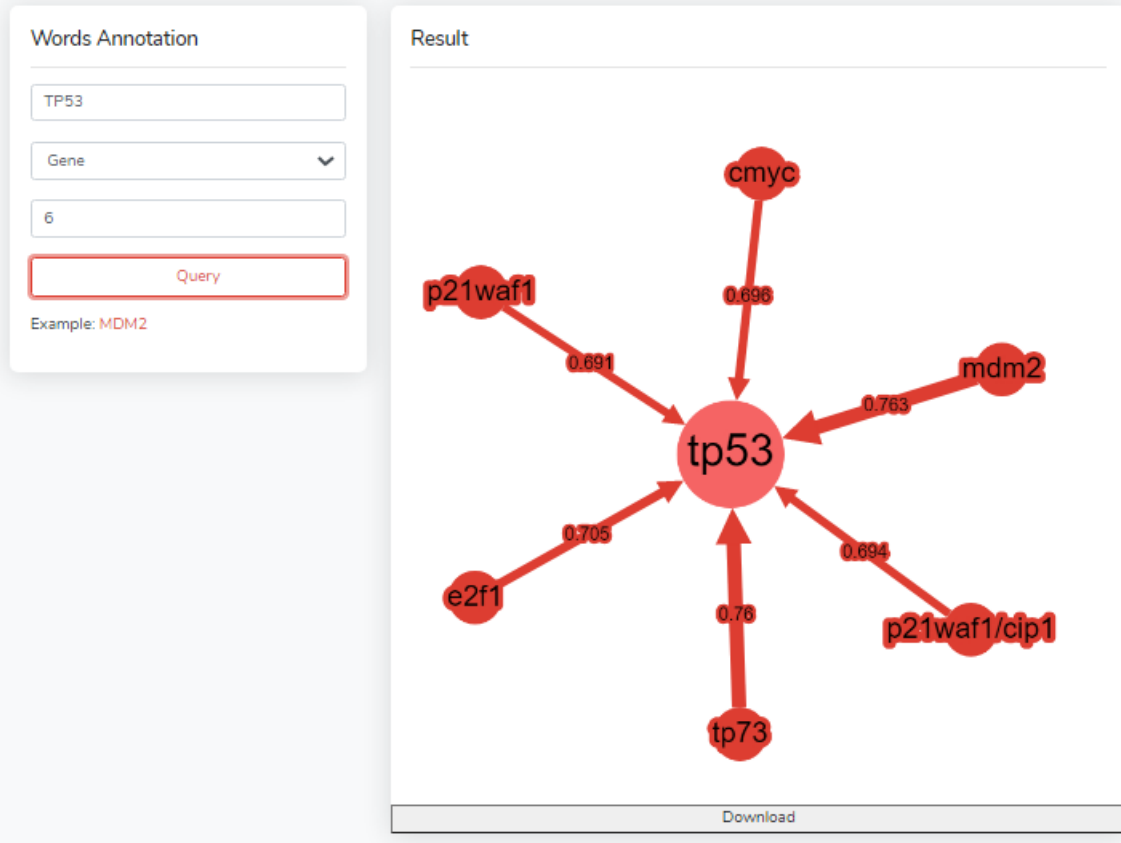


Figure 71. The first 6 nearest neighbors of the 'TP53' gene with 'gene' output type. The central node is the input word. The surrounding nodes are its neighbors. The links are the similarities.

- List of Nearest Words

This function takes as input a list of words and returns the combined nearest neighbors to all words (Figure 72) according to the following criteria:

- **Number of nearest neighbors = 0 and cosine similarity cutoff = 0**

The result is the combined similar words of the first 10 nearest neighbors of each word.

- **Cosine similarity cutoff = 0**

The result is the first combined nearest neighbors to each word according to the chosen number of nearest neighbors.

➤ **Number of nearest neighbors = 0**

The result is the first combined nearest neighbors to each word according to the chosen similarity value cutoff.

Example query

string: [/word\\_annotate\\_list\\_vector?words=BRCA1,BRCA2,TP53&size=5&cutoff=0.6](/word_annotate_list_vector?words=BRCA1,BRCA2,TP53&size=5&cutoff=0.6)

Example API

query: [/word\\_annotate\\_list\\_vector/api?words=BRCA1,BRCA2,TP53&size=5&cutoff=0.6](/word_annotate_list_vector/api?words=BRCA1,BRCA2,TP53&size=5&cutoff=0.6)

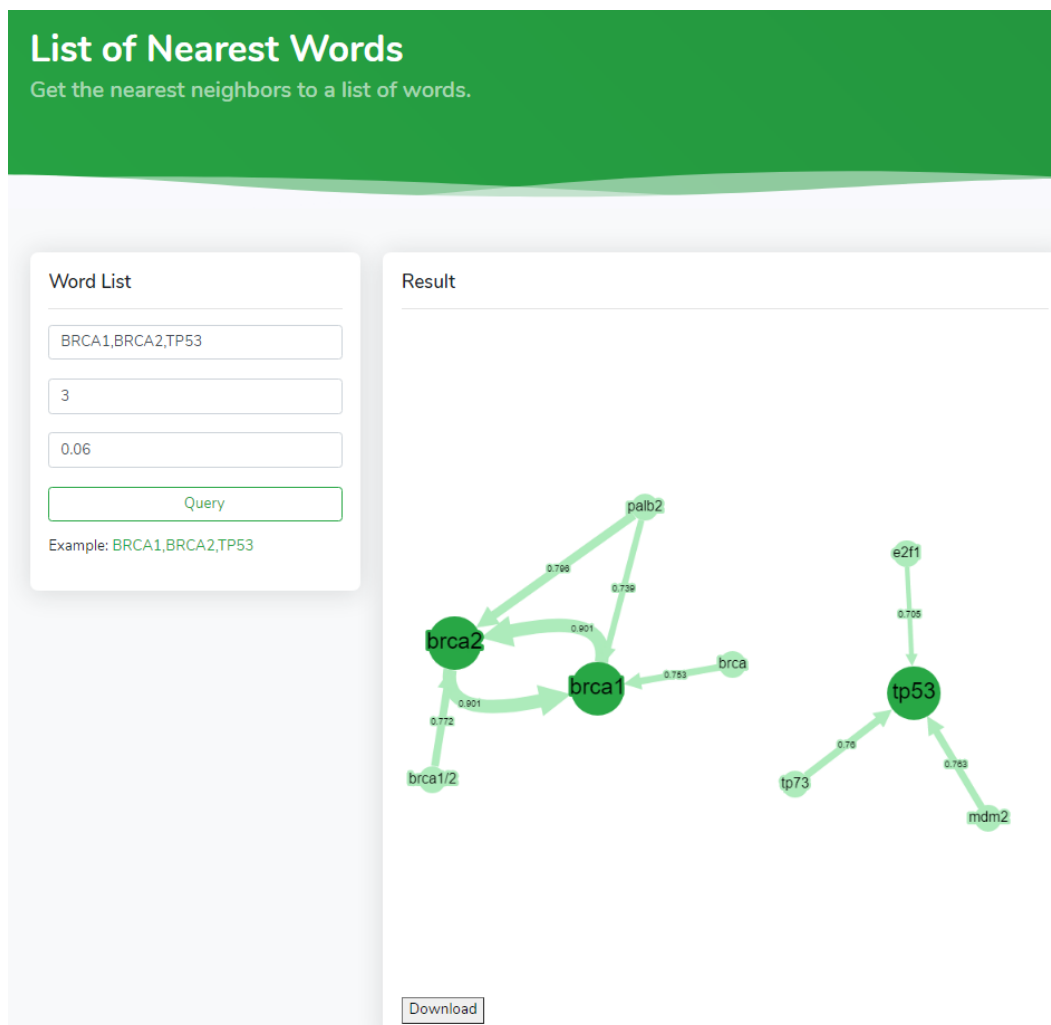


Figure 72. The combined nearest neighbors of the gene list 'BRCA1,BRCA2,TP53' with number of nearest neighbors=3 and a cosine similarity cutoff= '0.6'.



- Similarities between Words

This function returns the similarity between words within a list as network edges (Figure 73). Only the word pairs that have a similarity value  $\geq 0.4$  are returned. If a word is not in the vocabulary of the embedding, its similarity to other words cannot be calculated, so it won't be in the output list.

Example query string: [/list\\_edges\\_page?words=TP53,TP73,MDM2,MDM4,TP63](/list_edges_page?words=TP53,TP73,MDM2,MDM4,TP63)

Example API query: [/list\\_edges\\_page/api?words=TP53,TP73,MDM2,MDM4,TP63](/list_edges_page/api?words=TP53,TP73,MDM2,MDM4,TP63)

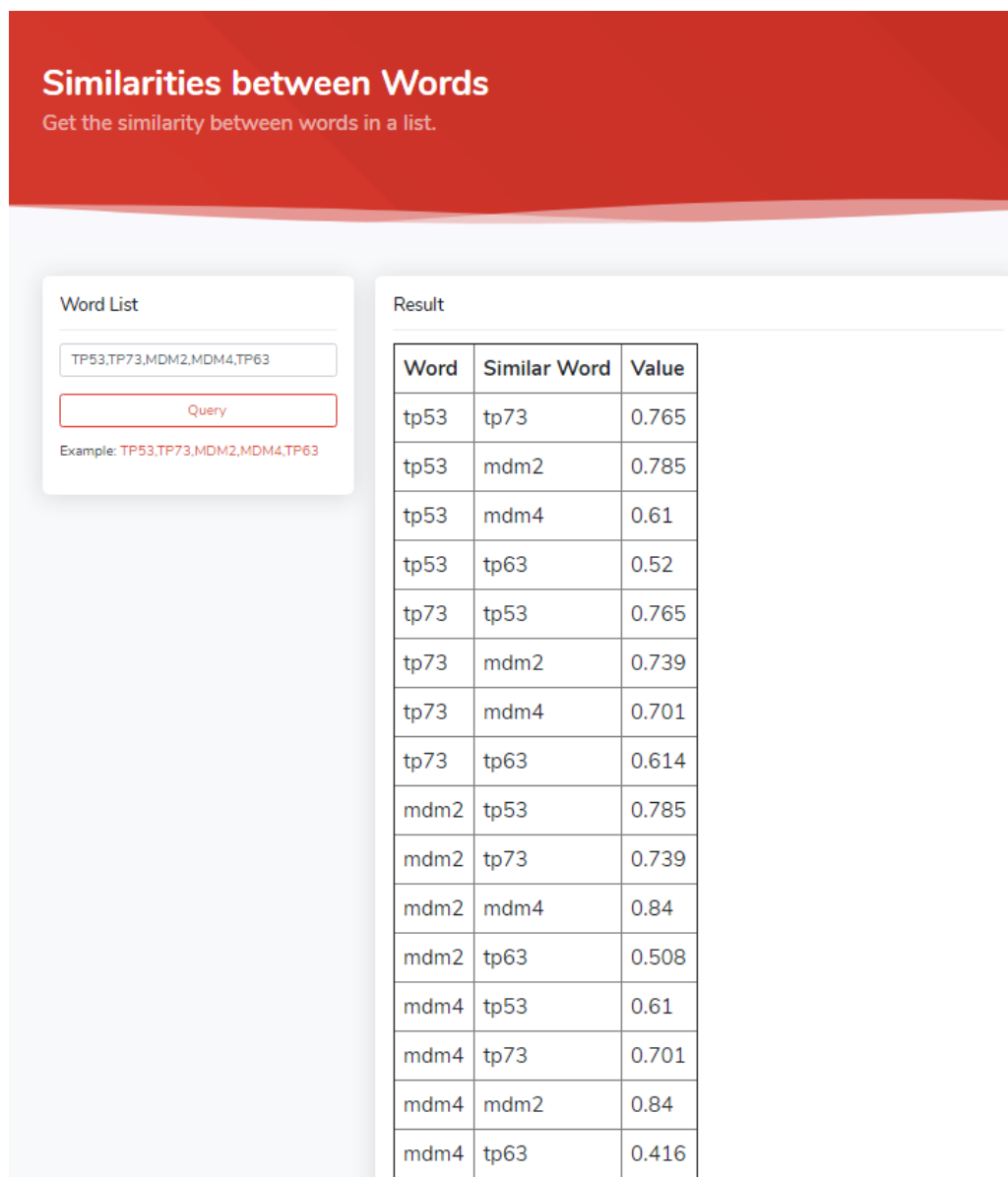


Figure 73. The similarities between entities in a gene list.

- Word Vector

This function returns the numerical vector generated by the trained model in 300 dimensions (Figures 74,75). Word vectors can be represented as numerical descriptors for biomedical concepts.

Example query string: [/word\\_vector\\_page?word=MDM2](/word_vector_page?word=MDM2)

Example API query: [/word\\_vector/api?word=MDM2](/word_vector/api?word=MDM2)

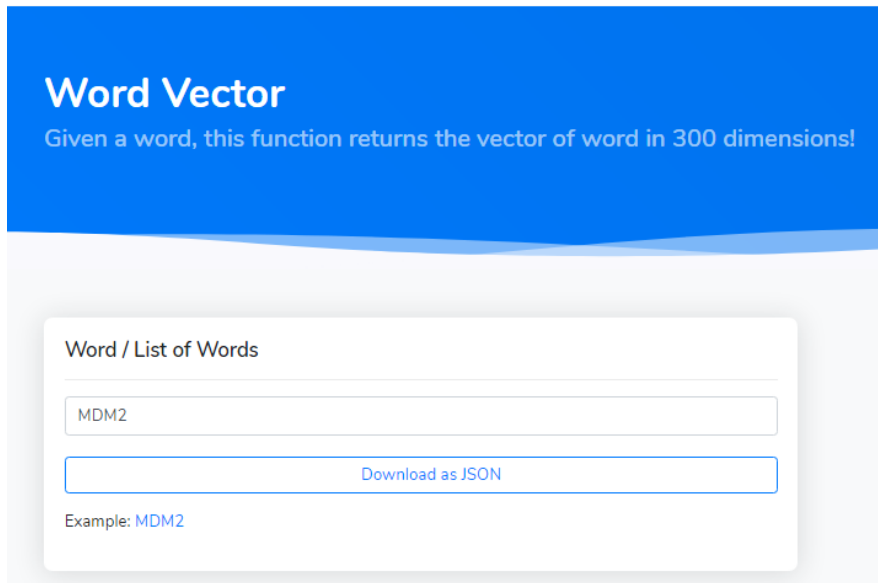


Figure 74. The 'Word Vector' function that returns a vector/vectors of a word/words in 300 dimensions.

```
]{
} ..... "mcm2" : : [
.....2.137035846710205 ,
.....0.26393139362335205 ,
.....0.49982649087905884 ,
.....-0.8965844511985779 ,
.....-1.3435344696044922 ,
.....0.7784188389778137 ,
.....-0.3687346875667572 ,
.....-1.9230365753173828 ,
.....-0.1743711233139038 ,
.....-1.4940974712371826 ,
.....0.6353750824928284 ,
.....-3.1998207569122314 ,
.....0.3328447639942169 ,
.....-0.07171355187892914 ,
.....0.34206390380859375 ,
.....3.2898244857788086 ,
.....-1.1093741655349731 ,
.....-0.13290449976921082 ,
.....
```

Figure 75. A glimpse of the numerical vector representation generated by the 'Word Vector' function.

All functions that have an input field for the number of nearest neighbors, if the number of nearest neighbors is not given, the first 10 nearest neighbors will be returned.

### **6.3.6. Computational Analysis Results**

To demonstrate the utility of our Word2Vec embedding in data analytical applications, we examined the agreement of cosine similarities between words according to their vector representations with information extracted from biomedical knowledge bases (see section 6.2.6). Each word in the resulting embedding is represented by a 300-dimensional numerical vector which is the default number of hidden layers used to train a Word2Vec model. As a result, pairs of genes with known interactions in the Reactome database showed higher cosine similarities than gene pairs without known interactions in the same database (Figure 76). Similarly, cosine similarities of drugs with overlapping target gene sets were on average higher than similarities between drugs without common target genes. Furthermore, cosine similarities within Reactome and TRANSPATH® pathways as well as within GO biological processes were increased compared to median cosine similarities of randomly sampled gene pairs (Figure 76). Regression curves estimated for the medians moreover revealed a correlation between the number of pathway or GO category members and the median similarity, with higher values for smaller gene sets. We think that gene pairs in smaller pathway networks or biological processes were more likely to correspond to direct molecular interactors which share a close functional context than in pathway or functional categories with a higher number of members and that the embedding in many cases indeed captured these relations. While disease-disease cosine similarities within HDO groups also revealed such a trend for groups with less than 25 members, median similarities within groups were often smaller than for randomly chosen disease pairs (Figure 76). Disease-disease relations captured by broader HDO groups, therefore, did not correspond well with vector presentations of the embedding. Better correspondence was observed for narrower disease groups but did not exceed similarities of random disease pairs.

Additionally, for drug-drug similarities based on gene groups, the estimated median in each group increased as the number of genes in a group increased (median: group 1= 0.192, group 2= 0.318, group 3= 0.396) (Figure 77).

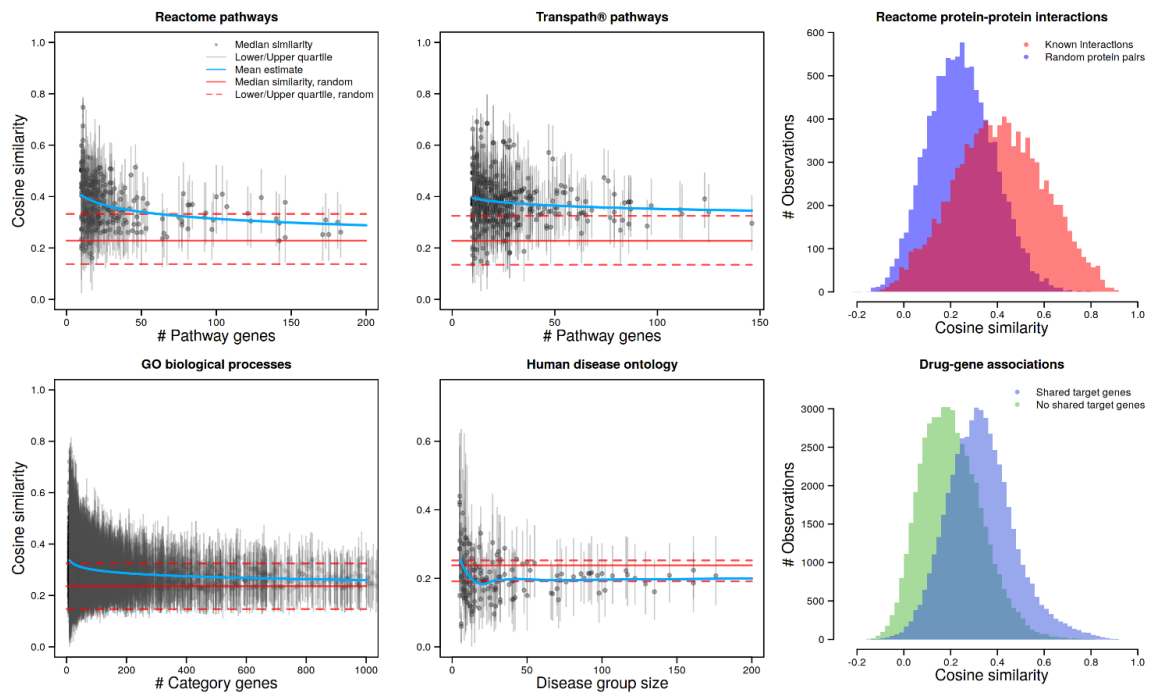


Figure 76. Validation of the Word2Vec embedding with existing knowledge from biomedical resources. Reactome pathways, TRANSPATH® pathways, GO biological processes and Human disease ontology present median cosine similarities as well as their lower and upper quartiles within groups of given number of members (genes or diseases, respectively) and for random samples (Material and methods). Mean estimates were computed by fitting the decay function to medians, with the exception of the Human disease ontology comparison where a non-parametric local regression (Loess) was applied. Reactome protein-protein interactions and drug-gene associations show histograms of genes with or without known PPIs and of drugs with or without shared target genes, respectively. (Figure used in [202])

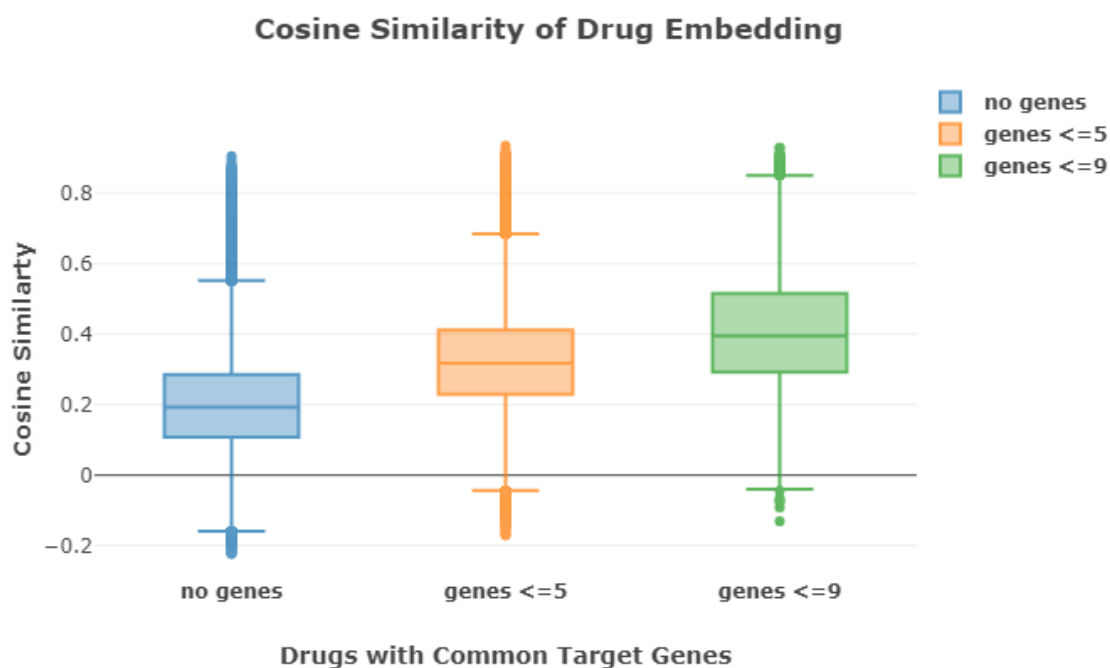


Figure 77. Boxplot of drug- drug cosine similarity distributions with shared genes of given number in DrugBank. Drug-drug groups were estimated by counting the number of shared genes between two drugs presented in the embedding. Group1 (no genes: lower quartile = 0.108, upper quartile = 0,286), group2 (genes  $\leq$  5: lower quartile = 0.229, upper quartile = 0,411), group3 (genes  $\leq$  9: lower quartile = 0.292, upper quartile = 0,516). (Figure used in [202])

### 6.3.7. Text Corpus Size Effect

We generated four embeddings trained with corpora of different sizes to examine the variability of cosine similarity values depending on the amount of training data. Figure 78 illustrates the results for selected terms of different types and their nearest neighbors. The cosine similarities are varying between the terms. For example, the similarity value between the “breast neoplasms” and “ovarian neoplasms” had increased slightly as follows: 0.851 (4M), 0.855 (8M), 0.861 (12M), 0.862 (~16M). Breast neoplasm is one of the most frequent diagnosed neoplasms reported in biomedical literature. Many studies have also reported similarities between breast and ovarian cancer since they share similar mutations (tumor suppressors). On the other hand, the similarity between "brca1" and "brca2" is almost the same in the four embeddings (0.898, 0.893, 0.891, 0.898 for 4M, 8M, 12M, and ~16M, respectively) with a very high similarity compared to the other nearest neighbors of "brca1". BRCA1 and BRCA2 genes are the most common genes defined in literature with certain mutations and lead to an increased

risk of breast and ovarian neoplasms. Similarly, for “schizophrenia” and “bipolar\_disorder”, their similarity has changed slightly (0.822, 0.825, 0.828, 0.829 for 4M, 8M, 12M, and ~16M, respectively). An overlap between schizophrenia and bipolar disorder has been commonly reported in the literature.

In contrast, the similarity between "eczema" and "atopy" had changed differently. It had decreased from 0.713 in the embedding with the corpus of size 4M to 0.669 in the one with 8M, to continue increasing again to 0.691 (12M) and 0.701 (~16M) while staying lower than their similarity in the embedding pre-trained with the smallest corpus.

Overall, we observed that the nearest neighbors of selected terms were assigned similarity values as well as a similar ranking that were varying slightly in the four embeddings for the majority of the selected terms. However, for common terms such as breast neoplasms, BRCA1, and schizophrenia and their nearest neighbors with which they tend to appear more frequently in biomedical literature, the similarity was more robust.

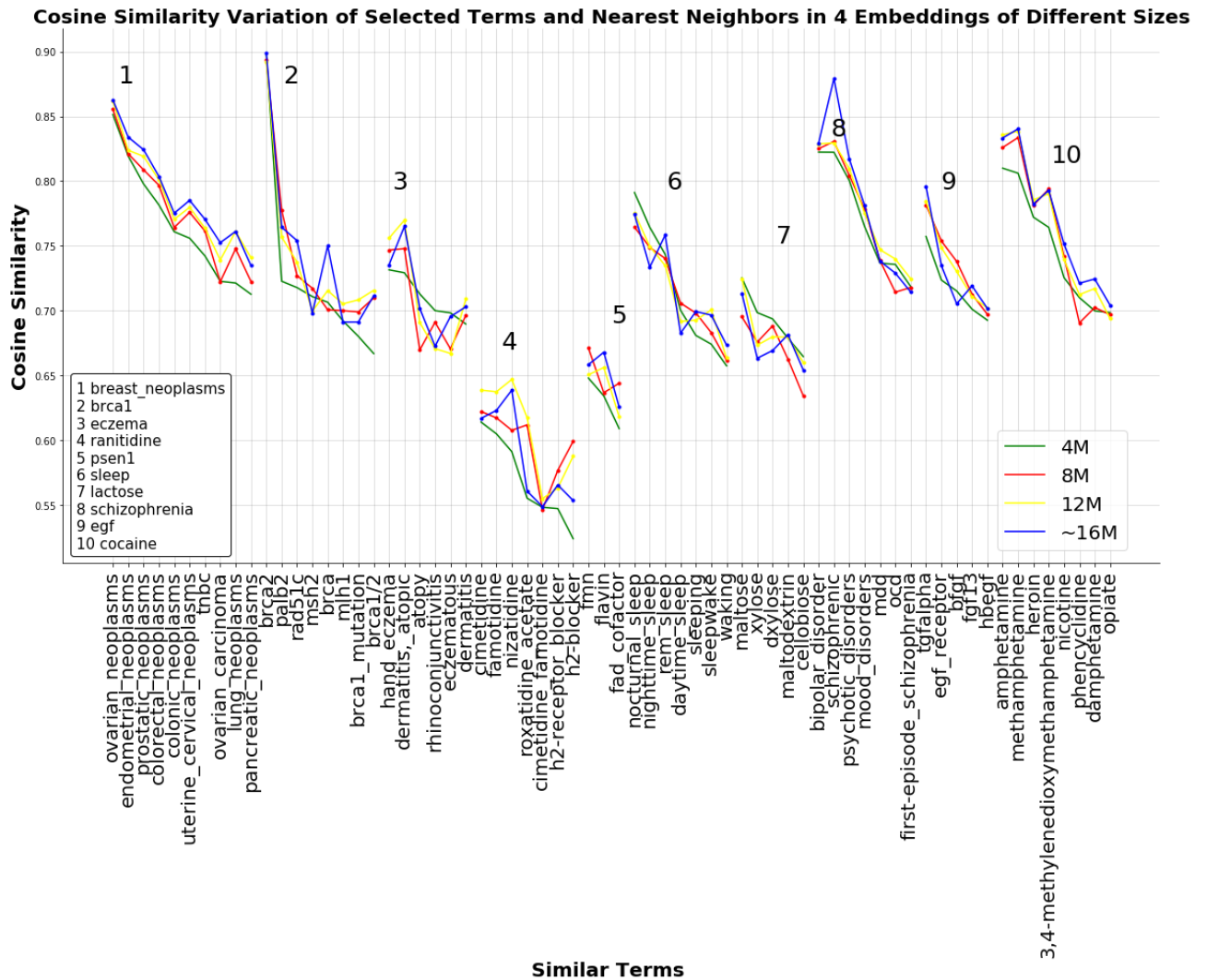


Figure 78. Assessment of similarities between selected terms and their nearest neighbors present in 4 embeddings. The selected terms are the genes *brca1*, *psen1* and *egf*, the medical terms breast neoplasms, eczema, sleep and schizophrenia, and the molecular compounds ranitidine, lactose, and cocaine; and their nearest neighbors present in 4 embeddings trained on a text corpus of different sizes. Each numbered subplot represents the cosine similarities between a selected term and its nearest neighbors from the four embeddings. Each plotted line represents one embedding and each dot on a line is the similarity between a selected term and one nearest neighbor. A point on the line is the similarity value between a selected term and one nearest neighbor. Nearest neighbors are arranged according to decreasing cosine similarity for the 4M corpus. (Figure used in [202])

### 6.3.8. Graph-CNN Performance Evaluation

Graph-CNN models were trained on the breast cancer data from section 2.3 to predict an occurrence of a metastatic event, utilizing different prior knowledge. The models with gene-gene networks derived from the embeddings showed the best performance compared to HPRD PPI, STRING-derived networks, BioBERT-derived network, and random network, in classifying patients into two groups, metastatic and non-metastatic. The networks were compared based on the similarity threshold and the number of vertices included. The architecture of Graph-CNN consists of 2 convolutional layers. Two convolutional layers were used with 32 convolutional filters. Maximum pooling of size 2 applies to both of the convolutional layers. Two fully connected layers have 512 and 128 nodes, consequently. The training was performed on 90% of data (872 patients) and 10% was saved as a test set (97 patients). The predicted patient data were classified into two groups: metastatic and non-metastatic.

Table 2 presents the performance of Graph-CNNs trained with the word2vec-embedding networks (`Embedding_net_v1` and `Embedding_net_v2`) and STRING derived networks incorporating the edge weights. For STRING, the edge weights are the scores computed based on text-mining techniques (see Materials and Methods). We didn't consider a weighted BioBERT-derived network since the minimal weight was already 0.938 for around 6000 vertices, which is not that much different from 1.0. We can see that `Embedding_net_v1` demonstrated a better performance than `Embedding_net_v2` for almost the same number of vertices and better than the text-mining-based STRING network.



Table 2. The results of how weighted underlying networks influence the performance of Graph CNN on the same data. The networks were compared based on the similarity threshold and number of vertices included. ‘Vertices’ are the vertices in the main connected component. ‘Similarity Threshold’ is the minimum weight value to keep connections between genes. AUC, Accuracy, and F1 weighted are the evaluation metrics used. ‘Epochs’ is the number of passing the entire dataset through the neural network.

Network	Vertices	Similarity Threshold	AUC %	Accuracy %	F1-weighted %	Epochs
Embedding_net_v1 (weighted)	6092	0.65	<b>83.09±0.97</b>	<b>76.67±1.14</b>	<b>76.45±1.14</b>	100
Embedding_net_v2 (weighted)	6086	0.68932	82.05±1.08	75.03±0.70	74.74±0.76	100
Embedding_net_v1 (weighted)	6775	0.63	82.53±1.46	75.62±1.72	75.29±1.73	100
Embedding_net_v2 (weighted)	6774	0.6774	81.87±1.39	75.33±1.22	75.16±1.28	40
STRING (text-mining) weighted	6840	0.744	81.76±1.95	75.97±1.99	75.63±2.02	100

In Table 3, we compared how unweighted network topologies influence the classifier’s performance depending on the similarity threshold and the number of vertices included. The baseline performance corresponds to HPRD PPI prior knowledge. STRING (combined) and BioBERT-based networks were considered only as unweighted since the weight thresholds to reach a comparable number of vertices were close to 1, 0.938, and 0.952, respectively.

The embedding networks have a threshold value allowing to change the strength of similarity between vertices. Change of threshold for Embedding\_net\_v1 from 0.63 to 0.65 increased the classification result in weighted and unweighted cases. We can also observe that for embedding networks, the incorporation of weight’s edges increased slightly, although not substantially, the classification performance. Meanwhile, STRING and BioBERT-based networks do not bring any improvements compared to HPRD PPI or the random network. Thus, Graph-CNNs showed the best results on our dataset, incorporating weighted Embedding\_net\_v1 with a threshold of 0.65.

Table 3. Influence of unweighted underlying networks on the performance of Graph CNN on the same data. The networks were compared based on the similarity threshold and number of vertices included. ‘Vertices’ are the vertices in the main connected component. ‘Similarity Threshold’ is the minimum weight value to keep connections between genes. AUC, Accuracy, and F1 weighted are the evaluation metrics used. ‘Epochs’ is the number of passing the entire dataset through the neural network.

Network	Vertices	Similarity Threshold	AUC %	Accuracy %	F1-weighted %	Epochs
HPRD	6888	-	82.57±1.25	76.07±1.30	75.82±1.33	100
Embedding_net_v1	6092	0.65	<b>83.02±1.09</b>	<b>76.38±1.44</b>	<b>76.14±1.47</b>	<b>40</b>
Embedding_net_v1	6775	0.63	82.41±1.20	76.03±1.53	75.69±1.53	100
Embedding_net_v2	6874	0.675	82.37±1.36	75.04±1.25	74.84±1.17	40
STRING (text mining)	6840	0.744	81.67±2.01	76.07±1.50	75.61±1.57	25
STRING (combined)	6862	0.938	81.77±1.17	74.62±1.56	74.33±1.64	100
BioBERT_v1.0_PubMed	6865	0.95245	82.26±1.27	74.81±1.53	74.61±1.50	40
Random network	6888	-	81.89±0.109	75.65±0.99	75.43±0.99	40

### 6.3.9. GLRP for delivering patient-specific subnetworks with Embedding-based Network

The established graph layer-wise relevance propagation (GLRP) method was applied to explain the predictions of Graph-CNNs trained on the breast cancer gene expression data introduced in section 6.2.7.1. We selected four breast cancer patients that were correctly predicted and visualized individualized PPI subnetworks delivered from the data set of the microarray (Table 4). Two subnetworks were assigned with luminal A (LumA), a common subtype. While the other two subnetworks were for patients with luminal B (LumB) and basal-like subtypes which are highly aggressive. The created PPI subnetworks are shown in Figures 79,80,81,82. We used the same web service technique [224] employed in [216] to visualize the subnetworks. The

node colors in the displayed subnetworks are based on 25% and 75% quantiles gene of expression levels with blue= low expression, yellow= normal expression, and red= high expression. The vertex size is based on the relevance of the scores within a subnetwork. The generated subnetworks for all correctly predicted patients are available and can be explored at: <http://mypathsem.bioinf.med.uni-goettingen.de/MetaRelSubNetVis/Embedded/>

The subnetworks from the embedding network showed explanations that were different from the ones provided with the HPRD PPI [216]. By comparing the visualized subnetworks, we can easily see that the underlying molecular network has affected the explanations of Graph-CNN.

Table 4. Four breast cancer patients that were correctly predicted. Two patients are with the luminal A (LumA) subtype. The other two are for patients with the basal-like and luminal B (LumB) subtypes. Metastatic event is the predicted event. (This table is based on a similar selection of subnetworks from [216]).

<b>Patient's ID</b>	<b>Breast cancer subtype</b>	<b>Metastatic predicted event</b>	<b>Time of Metastases, years</b>	<b>Last follow-up, years</b>
GSM615195	Basal	1	0.76	-
GSM615233	LumA	1	0.79	-
GSM150990	LumA	0	-	9.93
GSM282406	LumB	0	-	7.08

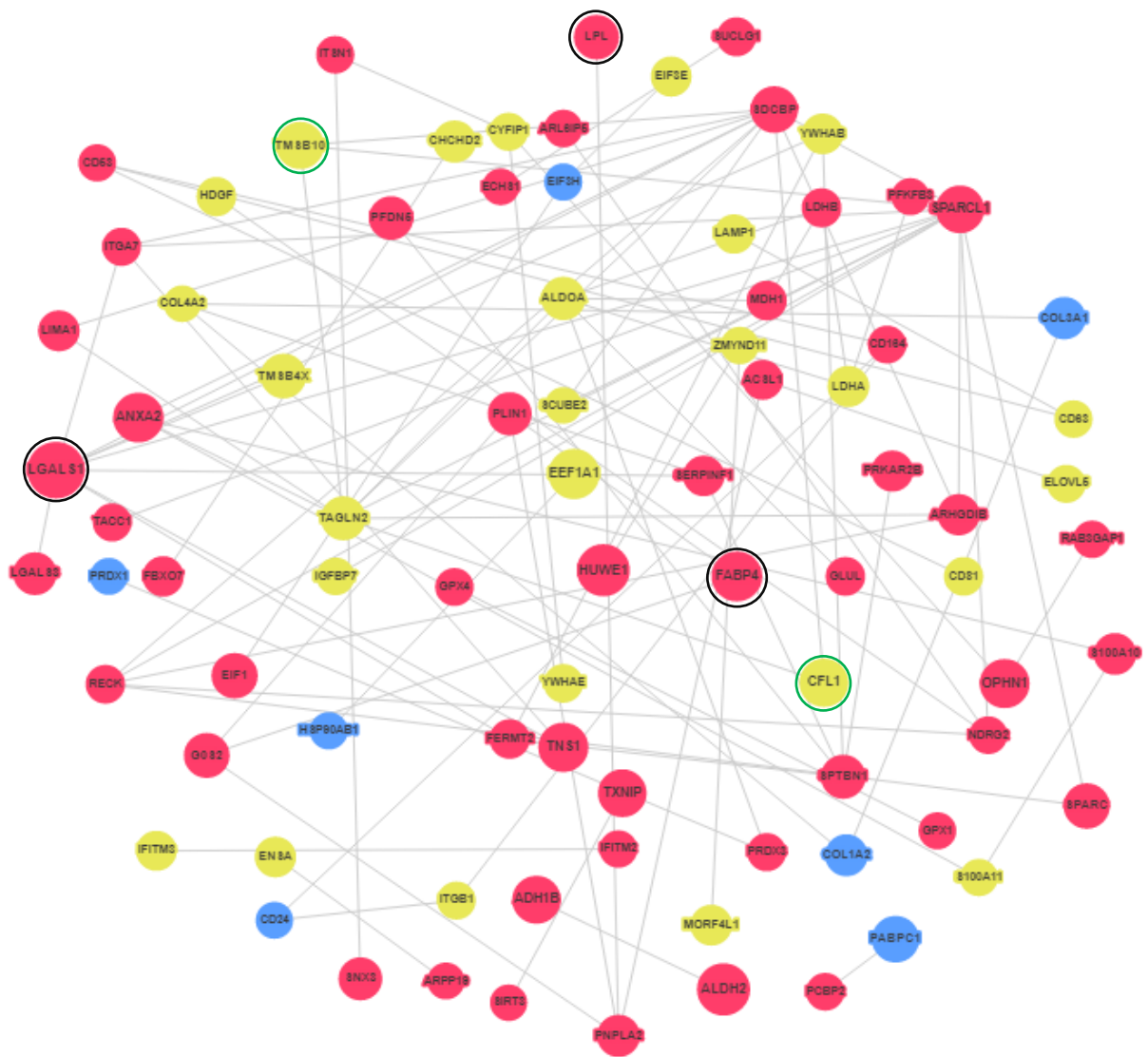


Figure 79. PPI subnetworks with the 140 most relevant genes for metastatic patient GSM615233 with Luminal A subtype.



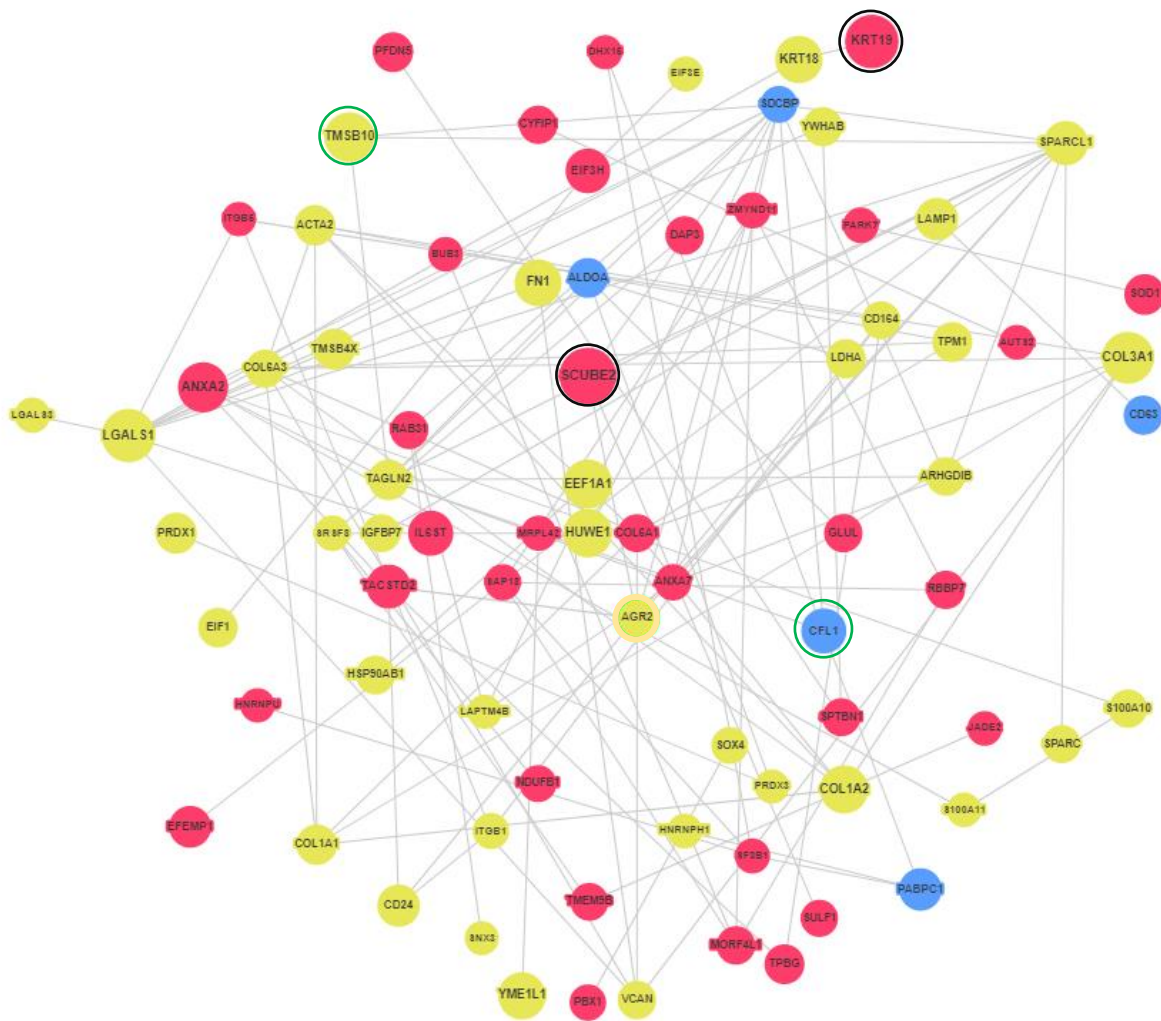


Figure 81. PPI subnetwork with the 140 most relevant genes for non-metastatic patient GSM150990 with Luminal A subtype.

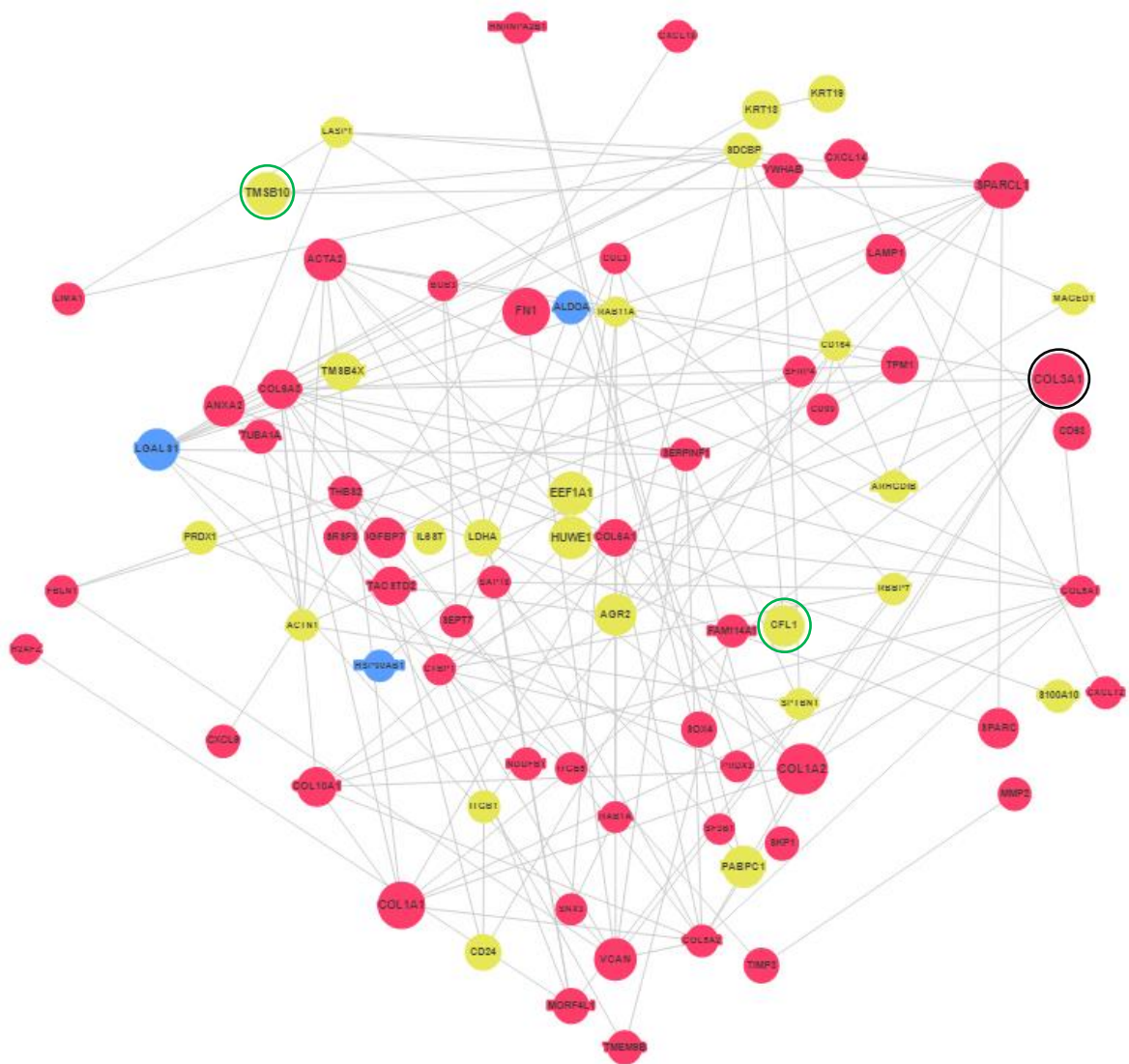


Figure 82. PPI subnetworks with the 140 most relevant genes for non-metastatic patient GSM282406 with Luminal B subtype.

The subnetworks generated by GLRP contained common oncogenes in the four patients, which may therefore be drivers that are common for the initiation and development of breast cancer. Examples are the actin-binding protein cofilin (CFL1) which regulates the invasiveness and motility of cancer cells [225] and the thymosin beta-10 (TMSB10) that plays a crucial role in the sequestration of G-actin and in the motility of breast cancer cells [226] (CFL1 and TMSB10 are highlighted by green in Figures 79,80,81,82).

In addition, a comparison of non-metastatic and metastatic patient subnetworks uncovered certain patient-specific genes which could provide useful knowledge on specific tumorigenesis pathways and help to identify therapeutic vulnerabilities for a specific patient. Each subnetwork contained specific cancer-related genes of high relevance in both metastatic and non-metastatic

patients. The metastatic patient subnetworks contained genes known to be implicated in the development of aggressive tumors. It was interesting that the metastatic subnetwork “GSM615233” (same subnetwork in [216]) included the genes: LPL and FABP4 (highlighted by black in Figure 79), which were shown to interact with CD36 to prevent apoptosis and to promote cell proliferation [227][228][229]. Also, LGALS1 (Galectin-1) is one of the most relevant genes in the patient metastatic subnetwork GSM615233 with Luminal A. Galectin-1 has been found to be active in multiple cancer cell invasion steps and metastasis. It regulates both cell migration and cell adhesion. It also interacts with extracellular matrix (ECM) molecules like fibronectin, laminin, integrin, and 90 K (MAC-2BP), which makes it play another vital role in cancer progression. By interacting with these molecules, galectin-1 regulates cell-ECM adhesion and modifies the aggregation and motility of cells. These interactions are essential steps in the metastasis and invasion of cancer cells [230][231].

Moreover, CHCHD2 and SDCBP are overexpressed and highly relevant in the metastatic patient subnetwork GSM615195 (highlighted by black in Figure 80). Overexpression of CHCHD2 in breast cancer patients is known to be related to distant metastasis and poor prognosis. Through upregulation of MMP2, CHCHD2 can induce the proliferative and migratory capacity in Docetaxel-resistant breast cancer cell lines [232]. SDCBP is an adapter protein that contains domains of PDZ. It leads to tumorigenicity and plays an important role in metastasis in many malignant tumors. It has also been documented that SDCBP is responsible for cell invasion, and the development of pseudopodia, which are associated with tumor metastasis [233]. In contrast, the non-metastatic subnetworks contained genes that are known to be related to tumor suppression. SCUBE2 and KRT19 are highly relevant and overexpressed in the non-metastatic patient subnetwork GSM150990 with luminal A (highlighted by black in Figure 81). SCUBE2 has a crucial role in suppressing the mobility and invasiveness of breast cancer cells. It promotes the increase of the development of epithelial E-cadherin which contains adhesive junctions in order to facilitate epithelial differentiation and epithelial-mesenchymal transition (EMT) reversal [234]. KRT19 has been shown to be involved in cancer progression regulation by acting as an oncogene or tumor suppressor gene [235]. It was shown that aggressiveness like cell proliferation and drug resistance, may be inhibited by the overexpression of KRT19 which suppresses the expression of the genes known to be associated with these phenotypes [235]. In GSM282406, COL3A1 has also shown high relevance while it is upregulated (highlighted by black in Figure 82). COL3A1 has been reported in breast cancer, with a poor prognosis association between COL3A1 and P4HA2 [236]. Col3 has also



been shown to suppress triple-negative breast cancer cell metastatic pathways, and tumor metastasis in mice [237].

## 6.4. Discussion

Biomedical and life sciences literature is increasing exponentially in volume while having an interdisciplinary nature. [71]. For many users, particularly clinicians and biomedical researchers, access to biomedical literature is fundamental. However, it is challenging to explore the huge amount of information available in natural language. Word embeddings have been a key technique in the biomedical domain used to represent words as vectors of real values in a predefined vector space. The evaluation of word embeddings has always been a continuous research question. In this work, we focused on demonstrating the validity and utility of biomedical word embeddings. We used the word2vec implementation in two approaches to investigate functional relations between biomedical entities. In the first approach, we trained the word2vec model by using disease vocabulary information as prior knowledge to generate disease-specific embeddings. Graphically displaying the embeddings of selected diseases and drugs in 3D space has helped to explore disease-drug associations and disease clusters. Firstly, it has been shown how diseases and drugs formed two respective clusters in the space by displaying them all together as well as within one particular category. We clearly showed the clusters by displaying diseases and drugs of the ‘neoplastic process system’ and ‘central nervous system’. On the other hand, we were able to depict related diseases and drugs. We identified 3 drugs related to Zollinger-Ellison syndrome and 7 diseases related to “Amphotericin B”. The number of identified terms was limited to the validated relations extracted from the medical resource “5 min consult”. Even though the number of identified relations was limited to two examples, it was able to validate relations between diseases and drugs from two perspectives (diseases related to drugs, drugs related to disease). Validated relations are useful to populate an empty ontology with entities and relationships. On the other hand, visualizing embedding can help capture information visually for just a limited number of entities. However, the larger is the embedding to visualize, the harder it would be to detect information.

For the second approach, we generated two representations of biomedical word embeddings using word2vec based on a text corpus consisting of PubMed abstracts. The two representations were generated for the purpose of comparison. One representation included synonymous terms that were substituted by their preferred terms. Validating semantic relationships between

biomedical entities represented in word embeddings has the potential to enhance the clarity of word embeddings and the interpretability of downstream tasks using them. We performed a computational analysis to validate similarities between biomedical entities namely, genes, diseases, and drugs using existing knowledge in biomedical databases. Comparisons showed that relations between entities such as known PPIs, common pathways and cellular functions, or narrower disease ontology groups correlated with higher vector cosine similarity. Gene pairs with known PPIs in Reactome have shown generally higher cosine similarities. Gene embeddings seem to be rich with semantic information about gene function. On the other hand, gene pairs with high cosine similarities shown without known interactions in Reactome or any other biomedical database would lead to new investigations of uncovering hidden functional relationships. In addition, gene pairs sharing common pathways in Reactome and TRANSPATH®, as well as common biological processes in GO, showed increased cosine similarities compared to the median of randomly sampled gene pairs. Moreover, similarities were increased with smaller group sizes which more likely represent direct molecular interactions. Disease pairs also showed increased cosine similarities within smaller HDO terms/groups e.g.  $\leq 20$  diseases, which likely represent more specific disease classes. However, disease embeddings did not correspond well on the basis of median random similarity. This is an interesting case to further investigate why semantic relations between diseases differ from the HDO, although it would contribute to new insights.

Corpus size effect assessment showed that similarities between selected terms were substantially affected by the corpus size. In general, we noticed that the first nearest neighbor for most terms was not strongly influenced by the corpus size even though it was not changing proportionally with the corpus size. The highest and strongest similarities were observed between "breast\_neoplasms" and "ovarian\_neoplasms" as well as between "brca1" and "brca2". This might be justified by the fact that these terms are very common in the present literature and words in each pair tend to occur more frequently in the same context. This could validate the ability to extract meaningful functional relationships between biomedical terms.

Additionally, in order to demonstrate the utility of the embedding in machine learning, we assumed that the similarities between biological entities might help to create networks of specific types. The results of Graph-CNN showed that the embedding-based networks are topologically meaningful. Weighted and unweighted Embedding\_net\_v1 allowed to increase the classification performance to predict the metastatic event in breast cancer according to the mean AUC, accuracy, and F1 measure. This can be explained by the fact that the integrated

information in the embedding is based on biological facts. The change of similarity threshold of edge weights from 0.63 to 0.65 led to increasing in performance, it can be due to the fact the network contained fewer vertices, and “weak genes” were filtered out. Random network-based demonstrated lower performance, although it is still to be investigated how simulated networks with different degree distributions would influence classification error rate. It was also shown that the model trained with the embedding\_net\_v1 network has performed better than with the embedding\_net\_v2. The former was produced from the embedding in which we replaced synonymous terms with their main terms. Such procedure has surely influenced the embedding information and in particular the semantic relations between terms. For example, considering the gene WNT4 and its nearest neighbor WNT7a, the cosine similarity between them has increased from 0.798 to 0.811, in Embedding\_v1 and Embedding\_v2 respectively. Although the increase in similarity was small, this has led to change it from being its third neighbor to becoming its first neighbor. Knowing that our examination was based only on gene-gene relations, it can, however, be extended to cover other types of relations e.g. disease-disease, gene-disease, etc. Moreover, our validation analysis was performed based on Embedding\_v2 while the similarities between biomedical entities can also be evaluated by checking the influence of Embedding\_v1.

The influence of embedding-based networks can be further examined by considering text-mining-based networks other than STRING and Bio-BERT. One could also derive networks from Bio-BERT using different hidden layers. For our Bio-BERT-derived network, the vectors of words were extracted from the last hidden layer. The BERT authors extracted vector combinations extracted from different layers. They tested the strategies of word-embedding by feeding those vector combinations to a BiLSTM (bidirectional long short-term memory) applied in a named entity recognition task and the resulting F1 scores were observed [213]. The best results on this task were provided by the concatenation of the last four layers. Therefore, for your particular application, it is best to compare different versions: the results may differ.

Our validation and evaluation analysis results will be published in a journal paper entitled: “Text Mining-Based Word Representations for Biomedical Data Analysis and Machine Learning Tasks”. The paper is based on our materials and methods in sections 6.2.6 and 6.2.7 and their sub-sections, as well as on our results in sections 6.3.3, 6.3.6, 6.3.7, and 6.3.8 including their sub-sections. A preprint of the paper is available under the following reference [202]:

- Alachram, Halima, Hryhorii Chereda, Tim Beißbarth, Edgar Wingender, and Philip Stegmaier. 2020. “Text Mining-Based Word Representations for Biomedical Data Analysis and Machine Learning Tasks.” *BioRxiv*, January, 2020.12.09.417733. <https://doi.org/10.1101/2020.12.09.417733>.

The individual patient-specific subnetworks generated from the Embedding\_v2 network have further demonstrated the biological utility of the embedding information. The subnetworks included common possible oncogenic drivers which suggest that they are capable of extracting important cancer pathways. They also included oncogenes observed in all four patient subnetworks and may be common drivers of the development and initiation of breast cancer. Moreover, both non-metastatic and metastatic patient subnetworks uncovered certain genes that are patient-specific, which could provide useful information on specific tumorigenesis pathways and therapeutic limitations in the respective patient. This would also demonstrate the biological utility of the embedding-based networks in molecular biology.

The developed “eBioMeCon” web service provides functions to explore similarities of biomedical concepts including the possibility to extract vertices using the function “Similarities between words”. Currently, the web service uses the embedding version we developed that excluded synonyms. However, the data can be extended by adding another embedding version and by providing the user with the option to select one of the embedding versions. Furthermore, specific embeddings can be generated for a specific domain such as an embedding using text articles that include information about a particular entity type like molecules or a particular disease type.

## 7 Conclusion

### 7.1. Summary

In this thesis, I developed applications that integrate, represent, and extract biomedical knowledge. The applications are useful to integrate patient clinical and genomics data with pathway knowledge in order to provide efficient data use in Systems Medicine. Enrichment analysis is an essential step to identify gene function and biological pathways related to differentially expressed genes in disease. The first application I developed is an enrichment analysis tool that utilizes a logistic regression-based method to identify predefined gene sets that are biologically related and enriched with genes that are differentially expressed. This approach was developed in a recent study as an R function. I adjusted the function by integrating the GO and Reactome categories annotated with Ensembl gene identifiers as predefined data sets. To use all the GO terms existing in the ontology graph which is essential for enrichment analysis tools that use GO annotations, I implemented functions by using the OBA service to access the ontology structure and to map genes annotated to descendants at the lowest level to all their ancestors. The tool is a Java-based application that can be easily used as a standalone application or it can be integrated into Java platforms.

My second application approach was to exploit molecular information available in existing resources that can be delivered to clinicians in order to help them modeling disease pathways. This can be established by linking clinical terms to concepts in biomedical databases. Therefore, I developed a lexical mapping module that works by comparing two concepts from two different resources using word-based and character-based metrics. The module includes a preprocessing phase that can be adjusted by the user to normalize texts. Using this module, I mapped ICD terms to disease concepts in the NCIT and the MeSH® vocabulary. To improve the results, I limited the information in the ICD and the NCIT ontologies to neoplasm concepts and I used other mapping strategies and mainly manual mapping as well as a structural strategy. The combination of these strategies was able to produce efficient mappings results. Furthermore, I exploited the NCIT ontology structure to develop functions to model disease pathways based on genes by using the relationships between the ontology concepts and their properties. I implemented these functions using the functionality of the OBA service. Using these functions, I developed a query that takes a disease name as input to return possible related pathways.

The scientific literature is a primary source from which such connections can be drawn and of special importance when aiming at newly discovered and experimentally confirmed relations. Approaches to extract novel knowledge from scientific literature using NLP and how the NLP results can be made accessible to downstream analysis are active research subjects. I used the word embedding technique, namely word2vec in two approaches to extract biological information. In the first approach, I used the Dis2vec, a modified word2vec model with a preprocessed text corpus to extract disease-drug associations. By visually representing the embedding results, I identified validated disease-drug association examples. The second approach was developing an embedding from a corpus I created and processed. To process the corpus, I applied different preprocessing strategies for comparison purposes. In addition, I developed a pipeline to process a text corpus and to generate word2vec embedding. Moreover, I developed functions to query information in the resulting embedding that can help to extract knowledge and to uncover hidden relationships. To facilitate the exploration of biomedical concepts in the embedding, I developed the eBioMeCon (embedding of biomedical concepts) web service. The service offers several ways to explore inferred contextual similarities between diseases, drugs, genes, and pathways through graphical and programming interfaces. To validate similarities between biomedical entities, I performed computational analyses using existing knowledge in biomedical databases. The analysis results showed that relations between entities such as known PPIs, common pathways and cellular functions, or narrower disease ontology groups correlated with higher vector cosine similarity. In addition, I assessed the effect of corpus size on the variability of similarities between selected terms and their nearest neighbors. To demonstrate the biological utility of the embedding information, I created gene-gene networks from two embedding versions and used them as prior knowledge to train Graph-Convolutional Neural Networks (CNNs) on breast cancer gene expression data to predict the occurrence of metastatic events. Performances of resulting models were compared to Graph-CNNs trained with protein-protein interaction networks or with networks derived using other word embedding algorithms. Graph-CNNs trained with word2vec-embedding-derived networks performed best for the metastatic event prediction task compared to PPI or other text mining-based networks. Word representations as produced by text mining algorithms like word2vec, therefore, capture biologically meaningful relations between entities. Our results demonstrated that high-throughput data interpretation can profit from the semantic relations to infer molecular interactions that play a role in disease development and therapeutic responses by using them as prior knowledge in machine learning tasks.

## 7.2. Outlooks

All the methods implemented in this manuscript are eligible to be extended and adjusted according to the intended purposes. For the LRpath enrichment tool, the categories in the datasets used to be tested can be extended by adding other annotation resources than GO and Reactome. Especially resources that include detailed information about pathways to promote studying the pathways that play a role in diseases. One more feature that can be added to the tool is the same feature presented in the original method, which is integrating LRpath results from multiple experiments and comparing the results in clustering analysis.

For the lexical mapping module, the application can be easily used by Java users. To make it more user-friendly, the module can be extended by developing an interactive interface and giving the user the ability to specify the threshold of the similarity score between two concepts. On the other hand, the implementation could be extended to tackle more specific cases during the preprocessing phase.

Representing biomedical concepts, which exist in literature, as numerical vectors was a fundamental task to explore relationships between entities. Neural network models always perform better when trained on larger datasets. The embedding I generated was based only on PubMed/MEDLINE® abstracts, however, it could be extended to include data from different biomedical text sources like PMC full text biomedical and life sciences articles, biomedical books, and other scientific literature. Another idea would be to develop embeddings that cover information about specific entity types such as proteins, or diseases. We have demonstrated that the embedding derived networks are biologically meaningful. Therefore, other networks of other entity types can also be created such as disease networks and drug networks. Such networks could also be tested in downstream applications or could be used to create knowledge bases. The developed web service based on the embedding data is already featured by functions that enable the exploration of biomedical concepts. However, it can be extended to include multiple embedding versions and give the user the choice to choose between them.

## Bibliography

- [1] R. Chen and M. Snyder, “Systems biology: personalized medicine for the future?,” *Curr. Opin. Pharmacol.*, vol. 12, no. 5, pp. 623–628, 2012.
- [2] E. W. Hinderer III, R. M. Flight, R. Dubey, J. N. MacLeod, and H. N. B. Moseley, “Advances in gene ontology utilization improve statistical power of annotation enrichment,” *PLoS One*, vol. 14, no. 8, 2019.
- [3] M. Ashburner *et al.*, “Gene ontology: tool for the unification of biology,” *Nat. Genet.*, vol. 25, no. 1, pp. 25–29, 2000.
- [4] A. Subramanian *et al.*, “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles,” *Proc. Natl. Acad. Sci.*, vol. 102, no. 43, pp. 15545–15550, 2005.
- [5] D. Na, H. Son, and J. Gsponer, “Categorizer: a tool to categorize genes into user-defined biological groups based on semantic similarity,” *BMC Genomics*, vol. 15, no. 1, p. 1091, 2014.
- [6] H. Tang *et al.*, “GOATOOLS: tools for gene ontology,” *Zenodo. doi*, vol. 10, 2015.
- [7] “map2slim - maps gene associations to a ‘slim’ ontology - metacpan.org.” [Online]. Available: <https://metacpan.org/pod/distribution/go-perl/scripts/map2slim>.
- [8] “Medical Subject Headings (MeSH) - WHSL Medical Subject Headings for PubMed Searching - LibGuides at University of the Witwatersrand.” [Online]. Available: <https://libguides.wits.ac.za/whsl-mesh>.
- [9] O. Bodenreider, “The unified medical language system (UMLS): integrating biomedical terminology,” *Nucleic Acids Res.*, vol. 32, no. suppl\_1, pp. D267–D270, 2004.
- [10] “ICD-10 Version:2016.” [Online]. Available: <https://icd.who.int/browse10/2016/en>.
- [11] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky, “GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles,” in *ISMB (supplement of bioinformatics)*, 2001, pp. 74–82.
- [12] H.-M. Müller, E. E. Kenny, and P. W. Sternberg, “Textpresso: an ontology-based information retrieval and extraction system for biological literature,” *PLoS Biol.*, vol.



- 2, no. 11, 2004.
- [13] M. C. Swain and J. M. Cole, “ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature,” *J. Chem. Inf. Model.*, vol. 56, no. 10, pp. 1894–1904, 2016.
- [14] S. Spangler *et al.*, “Automated hypothesis generation based on mining scientific literature,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1877–1886.
- [15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [16] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [17] B. Chiu, G. Crichton, A. Korhonen, and S. Pyysalo, “How to train good word embeddings for biomedical NLP,” in *Proceedings of the 15th workshop on biomedical natural language processing*, 2016, pp. 166–174.
- [18] Y. Wang *et al.*, “A comparison of word embeddings for the biomedical natural language processing,” *J. Biomed. Inform.*, vol. 87, pp. 12–20, 2018.
- [19] N. R. Smalheiser, A. M. Cohen, and G. Bonifield, “Unsupervised low-dimensional vector representations for words, phrases and text that are transparent, scalable, and produce similarity metrics that are not redundant with neural embeddings,” *J. Biomed. Inform.*, vol. 90, p. 103096, 2019.
- [20] A. R. Aronson and F.-M. Lang, “An overview of MetaMap: historical perspective and recent advances,” *J. Am. Med. Informatics Assoc.*, vol. 17, no. 3, pp. 229–236, 2010.
- [21] J. Kim, P. Pezik, and D. Rebholz-Schuhmann, “MedEvi: retrieving textual evidence of relations between biomedical concepts from Medline,” *Bioinformatics*, vol. 24, no. 11, pp. 1410–1412, 2008.
- [22] D. Rebholz-Schuhmann, M. Arregui, S. Gaudan, H. Kirsch, and A. Jimeno, “Text processing through Web services: calling Whatizit,” *Bioinformatics*, vol. 24, no. 2, pp. 296–298, 2008.

- [23] D. Campos, S. Matos, and J. L. Oliveira, “Gimli: open source and high-performance biomedical name recognition,” *BMC Bioinformatics*, vol. 14, no. 1, p. 54, 2013.
- [24] R. Hoffmann, “Using the iHOP information resource to mine the biomedical literature on genes, proteins, and chemical compounds,” *Curr. Protoc. Bioinforma.*, vol. 20, no. 1, pp. 1–16, 2007.
- [25] G. K. Savova *et al.*, “Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications,” *J. Am. Med. Informatics Assoc.*, vol. 17, no. 5, pp. 507–513, 2010.
- [26] C. Jonquet, N. H. Shah, and M. A. Musen, “The open biomedical annotator,” *Summit on Translat. Bioinforma.*, vol. 2009, p. 56, 2009.
- [27] J. Dönitz and E. Wingender, “The ontology-based answers (OBA) service: a connector for embedded usage of ontologies in applications,” *Front. Genet.*, vol. 3, p. 197, 2012.
- [28] K. Iida and I. Nishimura, “Gene expression profiling by DNA microarray technology,” *Crit. Rev. Oral Biol. Med.*, vol. 13, no. 1, pp. 35–50, 2002.
- [29] P. Tamayo *et al.*, “Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation,” *Proc. Natl. Acad. Sci.*, vol. 96, no. 6, pp. 2907–2912, 1999.
- [30] W. Dubitzky, M. Granzow, C. S. Downes, and D. Berrar, “Introduction to microarray data analysis,” in *A practical approach to microarray data analysis*, Springer, 2003, pp. 1–46.
- [31] “Microarray Technology: An introduction to DNA Microarray.” [Online]. Available: [http://www.premierbiosoft.com/tech\\_notes/microarray.html](http://www.premierbiosoft.com/tech_notes/microarray.html).
- [32] M. P. S. Brown *et al.*, “Support vector machine classification of microarray gene expression data,” *Univ. California, St. Cruz, Tech. Rep. UCSC-CRL-99-09*, 1999.
- [33] G. D. Bader, M. P. Cary, and C. Sander, “Pathguide: a pathway resource list,” *Nucleic Acids Res.*, vol. 34, no. suppl\_1, pp. D504–D506, 2006.
- [34] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, “KEGG: new perspectives on genomes, pathways, diseases and drugs,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D353–D361, 2017.

- [35] A. Fabregat *et al.*, “The reactome pathway knowledgebase,” *Nucleic Acids Res.*, vol. 46, no. D1, pp. D649–D655, 2018.
- [36] D. N. Slenter *et al.*, “WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research,” *Nucleic Acids Res.*, vol. 46, no. D1, pp. D661–D667, 2018.
- [37] C. F. Schaefer *et al.*, “PID: the pathway interaction database,” *Nucleic Acids Res.*, vol. 37, no. suppl\_1, pp. D674–D679, 2009.
- [38] E. G. Cerami *et al.*, “Pathway Commons, a web resource for biological pathway data,” *Nucleic Acids Res.*, vol. 39, no. suppl\_1, pp. D685–D690, 2010.
- [39] M. Krull, N. Voss, C. Choi, S. Pistor, A. Potapov, and E. Wingender, “TRANSPATH®: an integrated database on signal transduction and a tool for array analysis,” *Nucleic Acids Res.*, vol. 31, no. 1, pp. 97–100, 2003.
- [40] D. C. Kirouac, J. Saez-Rodriguez, J. Swantek, J. M. Burke, D. A. Lauffenburger, and P. K. Sorger, “Creating and analyzing pathway and protein interaction compendia for modelling signal transduction networks,” *BMC Syst. Biol.*, vol. 6, no. 1, p. 29, 2012.
- [41] D. Türei, T. Korcsmáros, and J. Saez-Rodriguez, “OmniPath: guidelines and gateway for literature-curated signaling pathway resources,” *Nat. Methods*, vol. 13, no. 12, p. 966, 2016.
- [42] H. Tipney and L. Hunter, “An introduction to effective use of enrichment analysis software,” *Hum. Genomics*, vol. 4, no. 3, pp. 1–5, 2010.
- [43] D. W. Huang, B. T. Sherman, and R. A. Lempicki, “Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists,” *Nucleic Acids Res.*, vol. 37, no. 1, pp. 1–13, 2009.
- [44] R. B. Burns and C. B. Dobson, “Chi-square,” in *Experimental Psychology*, Springer, 1981, pp. 223–242.
- [45] R. Routledge, “Fisher’s Exact Test,” *Encycl. Biostat.*, vol. 3, 2005.
- [46] J. J. Shuster, “Hypergeometric Distribution: Introduction,” *Wiley StatsRef Stat. Ref. Online*, 2014.
- [47] P. Khatri, S. Draghici, G. C. Ostermeier, and S. A. Krawetz, “Profiling gene

- expression using onto-express,” *Genomics*, vol. 79, no. 2, pp. 266–270, 2002.
- [48] T. Beißbarth and T. P. Speed, “GOstat: find statistically overrepresented Gene Ontologies within a group of genes,” *Bioinformatics*, vol. 20, no. 9, pp. 1464–1465, 2004.
- [49] X. Zhou and Z. Su, “EasyGO: Gene Ontology-based annotation and functional enrichment analysis tool for agronomical species,” *BMC Genomics*, vol. 8, no. 1, p. 246, 2007.
- [50] S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron, “Improved detection of overrepresentation of Gene-Ontology annotations with parent–child analysis,” *Bioinformatics*, vol. 23, no. 22, pp. 3024–3031, 2007.
- [51] A. Alexa, J. Rahnenführer, and T. Lengauer, “Improved scoring of functional groups from gene expression data by decorrelating GO graph structure,” *Bioinformatics*, vol. 22, no. 13, pp. 1600–1607, 2006.
- [52] R. Nogales-Cadenas *et al.*, “GeneCodis: interpreting gene lists through enrichment analysis and integration of diverse biological information,” *Nucleic Acids Res.*, vol. 37, no. suppl\_2, pp. W317–W322, 2009.
- [53] Y. Drier, M. Sheffer, and E. Domany, “Pathway-based personalized analysis of cancer,” *Proc. Natl. Acad. Sci.*, vol. 110, no. 16, pp. 6388–6393, 2013.
- [54] X.-M. Zhao *et al.*, “Identifying cancer-related microRNAs based on gene expression data,” *Bioinformatics*, vol. 31, no. 8, pp. 1226–1234, 2015.
- [55] J.-H. Lee *et al.*, “Integrative analysis of mutational and transcriptional profiles reveals driver mutations of metastatic breast cancers,” *Cell Discov.*, vol. 2, no. 1, pp. 1–14, 2016.
- [56] D. Yu, M. Kim, G. Xiao, and T. H. Hwang, “Review of biological network data and its applications,” *Genomics Inform.*, vol. 11, no. 4, p. 200, 2013.
- [57] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, “BioGRID: a general repository for interaction datasets,” *Nucleic Acids Res.*, vol. 34, no. suppl\_1, pp. D535–D539, 2006.
- [58] H.-W. Mewes *et al.*, “MIPS: a database for genomes and protein sequences,” *Nucleic*

*Acids Res.*, vol. 28, no. 1, pp. 37–40, 2000.

- [59] A. Franceschini *et al.*, “STRING v9. 1: protein-protein interaction networks, with increased coverage and integration,” *Nucleic Acids Res.*, vol. 41, no. D1, pp. D808–D815, 2012.
- [60] R. Hoehndorf, P. N. Schofield, and G. V Gkoutos, “The role of ontologies in biological and biomedical research: a functional perspective,” *Brief. Bioinform.*, vol. 16, no. 6, pp. 1069–1080, 2015.
- [61] “Social Research Glossary.” [Online]. Available: <http://www.qualityresearchinternational.com/socialresearch/ontology.htm>.
- [62] M. Salvadores, P. R. Alexander, M. A. Musen, and N. F. Noy, “BioPortal as a dataset of linked biomedical ontologies and terminologies in RDF,” *Semant. Web*, vol. 4, no. 3, pp. 277–284, 2013.
- [63] B. Smith *et al.*, “The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration,” *Nat. Biotechnol.*, vol. 25, no. 11, pp. 1251–1255, 2007.
- [64] “Marti Hearst: What Is Text Mining?” [Online]. Available: <http://people.ischool.berkeley.edu/~hearst/text-mining.html>.
- [65] L. Gong, “Application of Biomedical Text Mining,” *Artif. Intell. Emerg. Trends Appl.*, p. 417, 2018.
- [66] M. M. Islam, P. Hu, and Y. Wang, “Deep learning models for predicting phenotypic traits and diseases from omics datas,” *Artif. Intell. Emerg. Trends Appl.*, p. 333, 2018.
- [67] M. Islam, “Deep learning models for predicting phenotypic traits from omics data,” 2017.
- [68] “About PubMed by Year.” [Online]. Available: <https://esperr.github.io/pubmed-by-year/about.html>.
- [69] J. D. Saffer and V. L. Burnett, “Introduction to biomedical literature text mining: context and objectives,” in *Biomedical Literature Mining*, Springer, 2014, pp. 1–7.
- [70] “Home - PubMed - NCBI.” [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/>.
- [71] R. Khare, R. Leaman, and Z. Lu, “Accessing biomedical literature in the current

- information landscape,” in *Biomedical Literature Mining*, Springer, 2014, pp. 11–31.
- [72] H. Shatkay and M. Craven, *Mining the Biomedical Literature*. 2012.
- [73] Z. Lu, “PubMed and beyond: a survey of web tools for searching biomedical literature,” *Database*, vol. 2011, 2011.
- [74] N. C. for B. Information, “PubMed help,” 2007.
- [75] “MEDLINE, PubMed, and PMC (PubMed Central): How are they different?” [Online]. Available: <https://www.nlm.nih.gov/bsd/difference.html>.
- [76] R. I. Dogan, G. C. Murray, A. Névéol, and Z. Lu, “Understanding PubMed user search behavior through log analysis,” *Database*, vol. 2009, p. bap018, 2009.
- [77] “Citations Added to MEDLINE by Fiscal Year.” [Online]. Available: [https://www.nlm.nih.gov/bsd/stats/cit\\_added.html](https://www.nlm.nih.gov/bsd/stats/cit_added.html).
- [78] “Home - MeSH - NCBI.” [Online]. Available: <https://www.ncbi.nlm.nih.gov/mesh>.
- [79] M. A. Sartor, G. D. Leikauf, and M. Medvedovic, “LRpath: a logistic regression approach for identifying enriched biological groups in gene expression data,” *Bioinformatics*, vol. 25, no. 2, pp. 211–217, 2008.
- [80] “Home - geneXplain geneXplain.” [Online]. Available: <http://genexplain.com/>.
- [81] F. Cunningham *et al.*, “Ensembl 2019,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D745–D751, 2019.
- [82] D. Smedley *et al.*, “BioMart—biological queries made easy,” *BMC Genomics*, vol. 10, no. 1, p. 22, 2009.
- [83] “About the GO.” [Online]. Available: <http://geneontology.org/docs/introduction-to-go-resource/>.
- [84] “Introduction to GO annotations.”
- [85] T. Adamusiak *et al.*, “OntoCAT—simple ontology search and integration in Java, R and REST/JavaScript,” *BMC Bioinformatics*, vol. 12, no. 1, p. 218, 2011.
- [86] P. L. Whetzel *et al.*, “BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications,” *Nucleic Acids Res.*, vol. 39, no. suppl\_2, pp. W541–W545, 2011.

- [87] M. A. Musen, “The protégé project: a look back and a look forward,” *AI matters*, vol. 1, no. 4, pp. 4–12, 2015.
- [88] J. Dönitz *et al.*, “TrOn: an anatomical ontology for the beetle *Tribolium castaneum*,” *PLoS One*, vol. 8, no. 7, p. e70695, 2013.
- [89] J. Dönitz and E. Wingender, “EndoNet: an information resource about the intercellular signaling network,” *BMC Syst. Biol.*, vol. 8, no. 1, p. 49, 2014.
- [90] J. Dönitz *et al.*, “iBeetle-Base: a database for RNAi phenotypes in the red flour beetle *Tribolium castaneum*,” *Nucleic Acids Res.*, vol. 43, no. D1, pp. D720–D725, 2014.
- [91] I. Vastrik *et al.*, “Reactome: a knowledge base of biologic pathways and processes,” *Genome Biol.*, vol. 8, no. 3, p. R39, 2007.
- [92] D. Croft *et al.*, “The Reactome pathway knowledgebase,” *Nucleic Acids Res.*, vol. 42, no. D1, pp. D472–D477, 2013.
- [93] U. Consortium, “The universal protein resource (UniProt),” *Nucleic Acids Res.*, vol. 36, no. suppl\_1, pp. D190–D195, 2007.
- [94] “Home - Gene - NCBI.” [Online]. Available: <https://www.ncbi.nlm.nih.gov/gene/>.
- [95] J. Hastings *et al.*, “The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013,” *Nucleic Acids Res.*, vol. 41, no. D1, pp. D456–D463, 2012.
- [96] “Ensembl Stable IDs.” [Online]. Available: [https://www.ensembl.org/info/genome/stable\\_ids/index.html](https://www.ensembl.org/info/genome/stable_ids/index.html).
- [97] H.-G. Drost and J. Paszkowski, “Biomartr: genomic data retrieval with R,” *Bioinformatics*, vol. 33, no. 8, pp. 1216–1217, 2017.
- [98] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, “Entrez Gene: gene-centered information at NCBI,” *Nucleic Acids Res.*, vol. 39, no. suppl\_1, pp. D52–D57, 2010.
- [99] M. Kanehisa and S. Goto, “KEGG: kyoto encyclopedia of genes and genomes,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 27–30, 2000.
- [100] M. E. Edition, “MySQL: open source database. 2014,” URL <http://www.mysql.com/products/enterprise>.

- [101] J. Ooms, D. James, S. DebRoy, H. Wickham, and J. Horner, “RMySQL: Database interface and ‘MySQL’ driver for R: R package, version 0.10. 9.” 2016.
- [102] M. H. Satman, “RCaller: A software library for calling R from Java,” *J. Adv. Math. Comput. Sci.*, pp. 2188–2196, 2014.
- [103] “Enabling Open Innovation & Collaboration | The Eclipse Foundation.” [Online]. Available: <https://www.eclipse.org/>.
- [104] “Welcome to Apache NetBeans.” [Online]. Available: <http://netbeans.apache.org/>.
- [105] “Code reference information - Help | IntelliJ IDEA.” [Online]. Available: <https://www.jetbrains.com/help/idea/viewing-reference-information.html>.
- [106] “LRpath - Pathway Analysis using Logistic Regression.” [Online]. Available: <http://lrpath.ncibi.org/>.
- [107] B. Ofoghi, G. López-Campos, F. J. Martín-Sánchez, and K. Verspoor, “Mapping biomedical vocabularies: a semi-automated term matching approach,” in *ICIMTH*, 2014, pp. 16–19.
- [108] R. B. Altman and T. E. Klein, “Challenges for biomedical informatics and pharmacogenomics,” *Annu. Rev. Pharmacol. Toxicol.*, vol. 42, no. 1, pp. 113–133, 2002.
- [109] J. J. Cimino and E. H. Shortliffe, *Biomedical Informatics: Computer Applications in Health Care and Biomedicine (Health Informatics)*. Springer-Verlag, 2006.
- [110] E. Jiménez-Ruiz and B. C. Grau, “Logmap: Logic-based and scalable ontology matching,” in *International Semantic Web Conference*, 2011, pp. 273–288.
- [111] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, “The alignment API 4.0,” *Semant. Web*, vol. 2, no. 1, pp. 3–10, 2011.
- [112] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm, “Schema and ontology matching with COMA++,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 906–908.
- [113] “BioPortal Mappings - NCBO Wiki.” [Online]. Available: [https://www.bioontology.org/wiki/BioPortal\\_Mappings](https://www.bioontology.org/wiki/BioPortal_Mappings).
- [114] A. Ghazvinian, N. F. Noy, and M. A. Musen, “Creating mappings for ontologies in



- biomedicine: simple methods work,” in *AMIA Annual Symposium Proceedings*, 2009, vol. 2009, p. 198.
- [115] A. Zaeri and M. A. Nematbakhsh, “A Terminological Search Algorithm for Ontology Matching,” *Mod. Appl. Sci.*, vol. 6, no. 10, p. 37, 2012.
- [116] Y. Sun, L. Ma, and S. Wang, “A comparative evaluation of string similarity metrics for ontology alignment,” *J. Inf. & Computational Sci.*, vol. 12, no. 3, pp. 957–964, 2015.
- [117] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.,” 1990.
- [118] M. A. Jaro, “Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida,” *J. Am. Stat. Assoc.*, vol. 84, no. 406, pp. 414–420, 1989.
- [119] “tdebatty/java-string-similarity: Implementation of various string similarity and distance algorithms: Levenshtein, Jaro-winkler, n-Gram, Q-Gram, Jaccard index, Longest Common Subsequence edit distance, cosine similarity ...” [Online]. Available: <https://github.com/tdebatty/java-string-similarity>. [Accessed: 02-Jul-2021].
- [120] G. Stoilos, G. Stamou, and S. Kollias, “A string metric for ontology alignment,” in *International Semantic Web Conference*, 2005, pp. 624–637.
- [121] G. Kondrak, “N-gram similarity and distance,” in *International symposium on string processing and information retrieval*, 2005, pp. 115–126.
- [122] “n-gram - Wikipedia.” [Online]. Available: <https://en.wikipedia.org/wiki/N-gram>.
- [123] F. P. Miller, A. F. Vandome, and J. McBrewster, “Levenshtein distance: Information theory, computer science, string (computer science), string metric, damerau? Levenshtein distance, spell checker, hamming distance,” 2009.
- [124] L. Yujian and L. Bo, “A normalized Levenshtein distance metric,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1091–1095, 2007.
- [125] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, 1966, vol. 10, no. 8, pp. 707–710.
- [126] F. J. Damerau, “A technique for computer detection and correction of spelling errors,” *Commun. ACM*, vol. 7, no. 3, pp. 171–176, 1964.
- [127] D. Bakkelund, “An LCS-based string metric,” *Univ. Oslo*, 2009.

- [128] “python-string-similarity/README.md at master · luozhouyang/python-string-similarity · GitHub.” [Online]. Available: <https://github.com/luozhouyang/python-string-similarity/blob/master/README.md#longest-common-subsequence>.
- [129] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic cosine similarity,” in *The 7th International Student Conference on Advanced Science and Technology ICAST*, 2012, vol. 4, no. 1.
- [130] P. Jaccard, “Étude comparative de la distribution florale dans une portion des Alpes et du Jura,” 1901.
- [131] T. Sørensen *et al.*, “A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons,” 1948.
- [132] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text classification using string kernels,” *J. Mach. Learn. Res.*, vol. 2, no. Feb, pp. 419–444, 2002.
- [133] “String kernel - Wikipedia.”
- [134] “What Are ICD-10 Codes and How Do They Work?” [Online]. Available: <https://www.verywellhealth.com/icd-10-codes-and-how-do-they-work-1738471>.
- [135] “International Classification of Diseases, Version 10 - Summary | NCBO BioPortal.” [Online]. Available: <https://bioportal.bioontology.org/ontologies/ICD10>.
- [136] “The ICD10 Ontology is a formalization in OWL-DL of the International Classification of Diseases 10th edition, published by the World Health Organization (WHO) in 2004.” [Online]. Available: <https://dkm.fbk.eu/technologies/icd-10-ontology>.
- [137] “Medical Subject Headings - Home Page.” [Online]. Available: <https://www.nlm.nih.gov/mesh/meshhome.html>.
- [138] “About MEDLINE® and PubMed®: The Resources Guide.” [Online]. Available: <https://www.nlm.nih.gov/bsd/pmresources.html>.
- [139] D. Trieschnigg, P. Pezik, V. Lee, F. De Jong, W. Kraaij, and D. Rebholz-Schuhmann, “MeSH Up: effective MeSH text classification for improved document retrieval,”

- Bioinformatics*, vol. 25, no. 11, pp. 1412–1418, 2009.
- [140] R. R. Richter and T. M. Austin, “Using MeSH (medical subject headings) to enhance PubMed search strategies for evidence-based practice in physical therapy,” *Phys. Ther.*, vol. 92, no. 1, pp. 124–132, 2012.
- [141] “MeSH Browser.” [Online]. Available: <https://meshb.nlm.nih.gov/search>.
- [142] “Use of MeSH in Online Retrieval.” [Online]. Available: [https://www.nlm.nih.gov/mesh/intro\\_retrieval.html](https://www.nlm.nih.gov/mesh/intro_retrieval.html).
- [143] “NCI Thesaurus.” [Online]. Available: <https://ncit.nci.nih.gov/ncitbrowser/>.
- [144] “The OBO Foundry.” [Online]. Available: <http://obofoundry.org/>.
- [145] A. Kumar and B. Smith, “Oncology ontology in the NCI thesaurus,” in *Conference on Artificial Intelligence in Medicine in Europe*, 2005, pp. 213–220.
- [146] P. E. Hodges *et al.*, “Annotating the human proteome: the Human Proteome Survey Database (HumanPSD<sup>TM</sup>) and an in-depth target database for G protein-coupled receptors (GPCR-PD<sup>TM</sup>) from Incyte Genomics,” *Nucleic Acids Res.*, vol. 30, no. 1, pp. 137–141, 2002.
- [147] C. Choi *et al.*, “TRANSPATH®—a high quality database focused on signal transduction,” *Comp. Funct. Genomics*, vol. 5, no. 2, pp. 163–168, 2004.
- [148] Y. Li and P. Agarwal, “A pathway-based view of human diseases and disease relationships,” *PLoS One*, vol. 4, no. 2, 2009.
- [149] “ICD-10 Ontology | DKM.” [Online]. Available: <https://dkm.fbk.eu/technologies/icd-10-ontology>.
- [150] D. Nishimura, “BioCarta,” *Biotech Softw. Internet Rep. Comput. Softw. J. Sci.*, vol. 2, no. 3, pp. 117–120, 2001.
- [151] “Empowering App Development for Developers | Docker.” [Online]. Available: <https://www.docker.com/>.
- [152] D. Merkel, “Docker: lightweight linux containers for consistent development and deployment.”
- [153] S. S. Sahoo, O. Bodenreider, K. Zeng, and A. P. Sheth, “An experiment in integrating

- large biomedical knowledge resources with RDF: Application to associating genotype and phenotype information,” 2007.
- [154] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright, “NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information,” *J. Biomed. Inform.*, vol. 40, no. 1, pp. 30–43, 2007.
- [155] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, “Retrofitting word vectors to semantic lexicons,” *arXiv Prepr. arXiv1411.4166*, 2014.
- [156] Y. Cao, L. Huang, H. Ji, X. Chen, and J. Li, “Bridge text and knowledge by learning multi-prototype entity mention embedding,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1623–1633.
- [157] A. M. Turing, “Computing machinery and intelligence (1950),” *Essent. Turing Ideas that Gave Birth to Comput. Age. Ed. B. Jack Copeland. Oxford Oxford UP*, pp. 433–464, 2004.
- [158] H. Bunt and W. Black, *Abduction, belief and context in dialogue: studies in computational pragmatics*, vol. 1. John Benjamins Publishing, 2000.
- [159] “(Tutorial) Text ANALYTICS for Beginners using NLTK - DataCamp.” [Online]. Available: <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>.
- [160] M. Taboada, M. Meizoso, D. Martínez, and J. J. Des, “Using lexical, terminological and ontological resources for entity recognition tasks in the medical domain,” in *AIME Workshop on Knowledge Management for Health Care Procedures*, 2007, pp. 21–31.
- [161] C. Friedman, “Semantic text parsing for patient records,” in *Medical Informatics*, Springer, 2005, pp. 423–448.
- [162] Z. Gero and J. Ho, “PMCVec: Distributed phrase representation for biomedical text processing,” *J. Biomed. Informatics X*, vol. 3, p. 100047, 2019.
- [163] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv Prepr. arXiv1301.3781*, 2013.
- [164] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural

- architectures for named entity recognition,” *arXiv Prepr. arXiv1603.01360*, 2016.
- [165] Y. Wang *et al.*, “Clinical information extraction applications: a literature review,” *J. Biomed. Inform.*, vol. 77, pp. 34–49, 2018.
- [166] T. H. Nguyen and R. Grishman, “Employing word representations and regularization for domain adaptation of relation extraction,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 68–74.
- [167] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *arXiv Prepr. arXiv1406.1078*, 2014.
- [168] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, “Learning sentiment-specific word embedding for twitter sentiment classification,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1555–1565.
- [169] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, 2011, pp. 142–150.
- [170] J. Brownlee, *Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems*. Machine Learning Mastery, 2017.
- [171] “Unsupervised Feature Learning and Deep Learning Tutorial.” [Online]. Available: <http://deeplearning.stanford.edu/tutorial/supervised/SoftmaxRegression/>.
- [172] T. Mikolov, Q. V Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” *arXiv Prepr. arXiv1309.4168*, 2013.
- [173] “Word2vec: Google’s New Leap Forward on the Vectorized Representation of Words.” [Online]. Available: <https://medium.com/syncedreview/word2vec-googles-new-leap-forward-on-the-vectorized-representation-of-words-e8e505a31595>.
- [174] “A Gentle Introduction to Skip-gram (word2vec) Model — AllenNLP ver. — Real-World Natural Language Processing.” [Online]. Available: <http://www.realworldnlpbook.com/blog/gentle-introduction-to-skipgram-word2vec->

model-allennlp-ver.html.

- [175] “Semantic trees for training word embeddings with hierarchical softmax - Lateral.”
- [176] “Word2vec.” [Online]. Available: <https://devopedia.org/word2vec>.
- [177] F. Morin and Y. Bengio, “Hierarchical probabilistic neural network language model,” in *Aistats*, 2005, vol. 5, pp. 246–252.
- [178] J. Goodman, “Classes for fast maximum entropy training,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, 2001, vol. 1, pp. 561–564.
- [179] “Axhk97m/Words2vec: wordstovec report.” [Online]. Available: <https://github.com/Axhk97m/Words2vec>.
- [180] “Optimize Computational Efficiency of Skip-Gram with Negative Sampling | Pythonic Excursions.” [Online]. Available: [https://aegis4048.github.io/optimize\\_computational\\_efficiency\\_of\\_skip-gram\\_with\\_negative\\_sampling](https://aegis4048.github.io/optimize_computational_efficiency_of_skip-gram_with_negative_sampling).
- [181] S. Ghosh, P. Chakraborty, E. Cohn, J. S. Brownstein, and N. Ramakrishnan, “Characterizing diseases from unstructured text: A vocabulary driven word2vec approach,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1129–1138.
- [182] “Andrey Rzhetsky | Knowledge Lab | The University of Chicago.” [Online]. Available: [https://www.knowledgelab.org/people/detail/andrey\\_rzhetsky/](https://www.knowledgelab.org/people/detail/andrey_rzhetsky/).
- [183] “gensim: Introduction.” [Online]. Available: <https://radimrehurek.com/gensim/intro>.
- [184] “Natural Language Toolkit — NLTK 3.5b1 documentation.” [Online]. Available: <https://www.nltk.org/>.
- [185] P. M. Coates *et al.*, *Encyclopedia of Dietary Supplements (Online)*. CRC press, 2004.
- [186] M. E. Shils and M. Shike, *Modern nutrition in health and disease*. Lippincott Williams & Wilkins, 2006.
- [187] S. Povey, R. Lovering, E. Bruford, M. Wright, M. Lush, and H. Wain, “The HUGO gene nomenclature committee (HGNC),” *Hum. Genet.*, vol. 109, no. 6, pp. 678–680, 2001.

- [188] D. S. Wishart *et al.*, “DrugBank: a knowledgebase for drugs, drug actions and drug targets,” *Nucleic Acids Res.*, vol. 36, no. suppl\_1, pp. D901–D906, 2008.
- [189] D. S. Wishart *et al.*, “DrugBank: a comprehensive resource for in silico drug discovery and exploration,” *Nucleic Acids Res.*, vol. 34, no. suppl\_1, pp. D668–D672, 2006.
- [190] A. P. Davis *et al.*, “The comparative toxicogenomics database: update 2019,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D948–D954, 2019.
- [191] “A Beginner’s Guide to HTTP and REST.” [Online]. Available: <https://code.tutsplus.com/tutorials/a-beginners-guide-to-http-and-rest--net-16340>.
- [192] “How to Use an API with Python (Beginner’s Guide) [Python API Tutorial].” [Online]. Available: <https://rapidapi.com/blog/how-to-use-an-api-with-python/>.
- [193] “Understanding And Using REST APIs — Smashing Magazine.” [Online]. Available: <https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/>.
- [194] “CherryPy — A Minimalist Python Web Framework — CherryPy 18.5.1.dev7+g242dc04f.d20200224 documentation.” [Online]. Available: <https://docs.cherrypy.org/en/latest/index.html>.
- [195] “Welcome to web.py! (web.py).” [Online]. Available: <https://webpy.org/>.
- [196] “Bottle: Python Web Framework — Bottle 0.13-dev documentation.” [Online]. Available: <https://bottlepy.org/docs/dev/>.
- [197] “Welcome to Flask — Flask Documentation (1.1.x).” [Online]. Available: <https://flask.palletsprojects.com/en/1.1.x/>.
- [198] “The Web framework for perfectionists with deadlines | Django.” [Online]. Available: <https://www.djangoproject.com/>.
- [199] C.-H. Wei, A. Allot, R. Leaman, and Z. Lu, “PubTator central: automated concept annotation for biomedical full text articles,” *Nucleic Acids Res.*, vol. 47, no. W1, pp. W587–W593, 2019.
- [200] C.-H. Wei, H.-Y. Kao, and Z. Lu, “PubTator: a web-based text mining tool for assisting biocuration,” *Nucleic Acids Res.*, vol. 41, no. W1, pp. W518–W522, 2013.
- [201] P. Shannon *et al.*, “Cytoscape: a software environment for integrated models of biomolecular interaction networks,” *Genome Res.*, vol. 13, no. 11, pp. 2498–2504,

- 2003.
- [202] H. Alachram, H. Chereda, T. Beißbarth, E. Wingender, and P. Stegmaier, “Text mining-based word representations for biomedical data analysis and machine learning tasks,” *bioRxiv*, p. 2020.12.09.417733, Jan. 2020.
- [203] B. Jassal *et al.*, “The reactome pathway knowledgebase,” *Nucleic Acids Res.*, vol. 48, no. D1, pp. D498–D503, 2020.
- [204] L. M. Schriml *et al.*, “Human Disease Ontology 2018 update: classification, content and workflow expansion,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D955–D962, 2019.
- [205] M. Bayerlová *et al.*, “Ror2 signaling and its relevance in breast cancer progression,” *Front. Oncol.*, vol. 7, p. 135, 2017.
- [206] T. Barrett *et al.*, “NCBI GEO: archive for functional genomics data sets—update,” *Nucleic Acids Res.*, vol. 41, no. D1, pp. D991–D995, 2012.
- [207] R. A. Irizarry *et al.*, “Exploration, normalization, and summaries of high density oligonucleotide array probe level data,” *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.
- [208] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in neural information processing systems*, 2016, pp. 3844–3852.
- [209] H. Chereda, A. Bleckmann, F. Kramer, A. Leha, and T. Beissbarth, “Utilizing Molecular Network Information via Graph Convolutional Neural Networks to Predict Metastatic Event in Breast Cancer.,” *Stud. Health Technol. Inform.*, vol. 267, pp. 181–186, 2019.
- [210] “keras-team/keras: Deep Learning for humans.” [Online]. Available: <https://github.com/keras-team/keras>.
- [211] T. S. Keshava Prasad *et al.*, “Human protein reference database—2009 update,” *Nucleic Acids Res.*, vol. 37, no. suppl\_1, pp. D767–D772, 2009.
- [212] D. Szklarczyk *et al.*, “STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D607–D613, 2019.
- [213] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep



- bidirectional transformers for language understanding,” *arXiv Prepr. arXiv1810.04805*, 2018.
- [214] J. Lee *et al.*, “BioBERT: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.
- [215] “PyTorch-Transformers | PyTorch.” [Online]. Available: <https://pypi.org/project/pytorch-transformers/>.
- [216] H. Chereda *et al.*, “Explaining decisions of Graph Convolutional Neural Networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer,” *bioRxiv*, 2020.
- [217] K. Wang *et al.*, “NERO: A Biomedical Named-entity (Recognition) Ontology with a Large, Annotated Corpus Reveals Meaningful Associations Through Text Embedding,” *bioRxiv*, 2020.
- [218] “The 5-Minute Clinical Consult | Clinical Drug Information.” [Online]. Available: <https://www.wolterskluwercdi.com/lexicomp-online/5-minute-clinical-consult/>.
- [219] “WHOCC - ATC/DDD Index.” [Online]. Available: [https://www.whocc.no/atc\\_ddd\\_index/](https://www.whocc.no/atc_ddd_index/).
- [220] “Anatomical Therapeutic Chemical Classification System.” [Online]. Available: [https://en.wikipedia.org/wiki/Anatomical\\_Therapeutic\\_Chemical\\_Classification\\_System](https://en.wikipedia.org/wiki/Anatomical_Therapeutic_Chemical_Classification_System).
- [221] “Maya Software | Computer Animation & Modeling Software | Autodesk.” [Online]. Available: <https://www.autodesk.com/products/maya/overview?support=ADVANCED&plc=MAAYA&term=1-YEAR&quantity=1#>.
- [222] M. Abadi *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv Prepr. arXiv1603.04467*, 2016.
- [223] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [224] “frankkramer-lab/MetaRelSubNetVis.” [Online]. Available: <https://github.com/frankkramer-lab/MetaRelSubNetVis>. [Accessed: 02-Sep-2021].

- [225] W. Wang, R. Eddy, and J. Condeelis, “The cofilin pathway in breast cancer invasion and metastasis,” *Nat. Rev. Cancer*, vol. 7, no. 6, pp. 429–440, 2007.
- [226] X. Zhang *et al.*, “Thymosin beta 10 is a key regulator of tumorigenesis and metastasis and a novel serum marker in breast cancer,” *Breast Cancer Res.*, vol. 19, no. 1, pp. 1–15, 2017.
- [227] S. Guaita-Esteruelas *et al.*, “Exogenous FABP4 increases breast cancer cell proliferation and activates the expression of fatty acid transport proteins,” *Mol. Carcinog.*, vol. 56, no. 1, pp. 208–217, 2017.
- [228] Y. Liang *et al.*, “CD36 plays a critical role in proliferation, migration and tamoxifen-inhibited growth of ER-positive breast cancer cells,” *Oncogenesis*, vol. 7, no. 12, pp. 1–14, 2018.
- [229] N. B. Kuemmerle *et al.*, “Lipoprotein lipase links dietary fat to solid tumor cell proliferation,” *Mol. Cancer Ther.*, vol. 10, no. 3, pp. 427–436, 2011.
- [230] J. M. Cousin and M. J. Cloninger, “The role of galectin-1 in cancer progression, and synthetic multivalent systems for the study of galectin-1,” *Int. J. Mol. Sci.*, vol. 17, no. 9, p. 1566, 2016.
- [231] E. Jung *et al.*, “Galectin-1 expression in cancer-associated stromal cells correlates tumor invasiveness and tumor progression in breast cancer,” *Int. J. cancer*, vol. 120, no. 11, pp. 2331–2338, 2007.
- [232] L. Ma, L. H. Zheng, D. G. Zhang, and Z. M. Fan, “CHCHD2 decreases docetaxel sensitivity in breast cancer via activating MMP2,” *Eur. Rev. Med. Pharmacol. Sci.*, vol. 24, no. 11, pp. 6426–6433, 2020.
- [233] X.-L. Qian *et al.*, “Syndecan binding protein (SDCBP) is overexpressed in estrogen receptor negative breast cancers, and is a potential promoter for tumor proliferation,” *PLoS One*, vol. 8, no. 3, p. e60046, 2013.
- [234] Y.-C. Lin, Y.-C. Lee, L.-H. Li, C.-J. Cheng, and R.-B. Yang, “Tumor suppressor SCUBE2 inhibits breast-cancer cell migration and invasion through the reversal of epithelial–mesenchymal transition,” *J. Cell Sci.*, vol. 127, no. 1, pp. 85–100, 2014.
- [235] S. K. Saha, K. Kim, G.-M. Yang, H. Y. Choi, and S.-G. Cho, “Cytokeratin 19 (KRT19) has a role in the reprogramming of cancer stem cell-like cells to less

aggressive and more drug-sensitive cells,” *Int. J. Mol. Sci.*, vol. 19, no. 5, p. 1423, 2018.

- [236] H. Engqvist *et al.*, “Immunohistochemical validation of COL3A1, GPR158 and PITHD1 as prognostic biomarkers in early-stage ovarian carcinomas,” *BMC Cancer*, vol. 19, no. 1, p. 928, 2019.
- [237] B. K. Brisson *et al.*, “Type III collagen directs stromal organization and limits metastasis in a murine model of breast cancer,” *Am. J. Pathol.*, vol. 185, no. 5, pp. 1471–1486, 2015.