ID N. 2

UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

DEPARTMENT OF ENGINEERING

UNIVERSITÀ DI PAVIA

DEPARTMENT OF ELECTRICAL, COMPUTER AND BIOMEDICAL ENGINEERING

Italian National Ph.D. in Artificial Intelligence
Health and Life Sciences
XXXVII Cycle

Integrate Pre-existing Knowledge in Biomedical Data Analysis with Graph Representation Learning

Supervisors
Arianna Dagliati
Riccardo Bellazzi

Candidate
Giuseppe Albi

November, 2025

Abstract

Early Artificial Intelligence (AI) systems encoded domain-specific knowledge using knowledge representation techniques, such as expert systems. In contrast, modern AI paradigms like Deep Learning have thrived due to the abundance of data and computational resources, shifting toward a data-driven approach to decision-making. However, relying solely on this kind of approach can pose risks in certain fields, particularly in medicine and biomedical research where expert knowledge plays a crucial role. Life sciences are inherently guided by curated, domain-specific expertise, which is empirically integrated with experimental data.

A promising direction is offered by the adoption of hybrid AI approaches that combine structured knowledge, such as medical understanding of specific phenotypes or known relationships among patient variables, with data-driven models. This integration can bridge the gap between structured knowledge representation and the flexibility of learning from data. The feasibility of hybrid methods is supported by the growing availability of structured biomedical knowledge, including knowledge bases (KBs), terminologies, and domain-specific bioinformatics repositories focused on areas such as precision medicine and pharmacology.

Graph Representation Learning (GRL) has gained increasing attention in biology and medicine due to its ability to model complex relationships using graph structures. Graphs offer an efficient way to represent entities, their interconnections, and associated attributes as informative signals. GRL is especially valuable because it allows for the integration of heterogeneous data sources, such as real-world patient data or public available data repository, with pre-existing biomedical knowledge. Knowledge Graphs (KGs), in particular, are effective in this context, as they harmonize diverse biomedical knowledge resources within a unified graph-based framework, facilitating comprehensive analysis in specialized domains.

The objective of this thesis is to proposes AI frameworks based on GRL paradigms and integrating pre-existing knowledge, to address the analysis of biomedical data from the INTESTRAT- CAD projects and computational toxicology. A Methodological Background reported in Chapter 2, starts by describing the graph notation and the Machine Learning (ML) tasks formulations on graph structures. Then, the chapter proceeds by introducing GRL methods including traditional graph statistics, manifold learning, Topological Data

Analysis (TDA) and graph embedding models learned with Neural Networks (NN) architectures. In the following chapters three case studies are reported, each addressing a specific biomedical task by adopting a suitable GRL paradigm, and identifying a thesis's aim: stratification and computational phenotyping (Aim 1), prediction of coronary artery stenosis for risk stratification (Aim 2), and small molecules toxicology prediction (Aim 3). Specifically, Aim 1 and Aim 2 use real-world data from the INTESTRAT-CAD project, with a patients' population belonging to the Epifania trial, while Aim 3 uses Tox21, a public available toxicology repository. In addition, Aim 1 considers as pre-existing knowledge the initial phenotypic medical definition assigned to the population in study, while Aim 2 and Aim 3 share similar sources of biomedical knowledge, by leveraging semantic relations from KGs.

Chapter 3 describes Aim 1, that deals with the Computational Phenotyping of Coronary Artery Disease (CAD) patients, on the basis of clinical data, and the domain medical knowledge defined as the initial CAD severity, assigned to the patients when enrolled in the study. A cohort of 725 patients is used to create a dataset made by clinical variables such as demographics, medical history, laboratory exams and drug prescriptions, and comprising the initial CAD definition assigned by the clinicians. The contribution is a TDA-based framework for semi-supervised computational phenotyping, where the Mapper hyper-parameters tuning is guided by the initial CAD label, and the characterization of the new subgroups is performed with the most discriminative features extracted from ML predictive models.

Chapter 4 reports Aim 2, dealing with the developing of a predictive risk model for CAD, by combining omics and clinical dataset variables with PrimeKG, a precision medicine-oriented KG. A different cohort from the previous Aim is considered, and consisting in 723 patients, with a wider set of clinical variables and the RNA-sequencing (RNA-seq) data. By first mapping the dataset features with PrimeKG nodes, and then using Knowledge Graph Embedding (KGE) models to learn the KG entities representations, this study shows how to contextualize real-word data with pre-existing medical knowledge, in order to create a new patient representation to be used for Coronary Artery Stenosis prediction.

Chapter 5 reports Aim 3, which objectives is to augment a toxicity prediction model with semantic knowledge deriving from ComptoxAI, a computational toxicology KG. Specifically, Graph Neural Networks (GNNs) are adopted to learn small molecules representation, by leveraging their 2-dimensional (2D) structure, atoms and edges attributes. In addition, this representation is updated with the knowledge between chemicals, genes and Tox21 assays, extracted from ComptoxAI, all combined to create a computational toxicology predictive pipeline.

Finally, the main motivation behind the thesis are highlighted, and the major findings with and possible future developments are discussed in Chapter 6.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Knowledge in the context of Artificial Intelligence in medicine

1.1.1 The evolution of medical knowledge in AI

A traditional definition of *knowledge* from the field of knowledge engineering is given by first defining the difference with *data* and *information*. While data are essentially uninterpreted signals, and information is data equipped with a particular meaning, the knowledge is *context-dependent* and represent the entire body of data and information to be brought into practical use in task action [191]. This definition is linked to the way firsts applications of Artificial Intelligence (AI) from the 1970s, are created from knowledge engineer, a figure that is responsible to elicit, or externalize, the knowledge from a specific domain, that is usually tacit, or implicit, from knowledge provider such as expert in the field [151]. The knowledge acquired is made available to the possible needs of a user, such as decision maker, through information systems called *knowledge-* or *expert-systems*, where the set of facts that constitute the knowledge on the specific domain resides in special informatics repository generally named Knowledge Bases (KBs) [238].

One of the most famous example of expert system, that was first proposed in the medical field, is MYCIN, which encapsulate knowledge trough logical rules, and addresses the diagnosis and treatment of infectious diseases [195]. This kind of first AI tool privilege a *deductive* approach to face decision making, in which the knowledge is used to create hypothesis that drive conclusions, without generating explicitly new knowledge [92].

With the rapid development of information technology that generate massive amounts of data globally, the boundaries between data, information and knowledge seems nowadays

thinner in the biomedical field. Indeed, modern medical AI applications such as foundational models, leverage paradigm closer to an *inductive* approach, e.g. Deep Learning (DL), to extract information from a big amount of unstructured data, especially if coming from different types, or modalities, to be able to solve different tasks but from the same medical domain of interest [22].

1.1.2 Why adopting a hybrid AI approach in medicine?

Many types of research question in medicine can be addressed even before understanding the direct cause of a specific therapeutic or pathologic mechanism. For example, the identification of clusters of patients can empirically reveal new taxonomy of the disease, suggests insight about the selection of the treatment strategies and predicting outcomes, generating insight about the underlying mechanism of the disease itself [115].

However, working exclusively by adopting a data-driven approach is recognized as to be at strong risk, since accumulating data without knowledge extraction can led to poor formalization of novel discoveries. Albeit, an approach that rely only on pre-defined guideline may strongly suffer from a lack of flexibility [159]. These considerations are especially valid for the biomedical research, a field driven by expert knowledge, and that combines insight derived empirically from existing data with the ones based on theory and experiment [115].

The adoption of hybrid AI approach in the biomedical research ,that combines knowledge with data-driven methods, is supported by the abundance of structured biomedical knowledge, available through KBs that are specialized on different biological domains, such as the pharmaceutical [113, 232], bioinformatics [105] and molecular biology [207, 11], clinical terminology [21, 198], and initiatives from the scientific community, such as the creation of publicly datasets [211].

1.2 Graph Representation Learning in biomedical research

1.2.1 Representation Learning

Representation learning (RL) is a sub-field of Machine Learning (ML) which object is to learn an efficient representation of the input data to be subsequently used for a downstream task, such as predictions or patterns discovery [18]. While RL has gained popularity thanks to the advent of DL [80], it can be generally considered as the application of computational methods suitable for the *geometrical* structure of the input, offering efficient alternative to

the traditional methods based on domain statistics.

For example, in Natural Language Processing (NLP), that is the field that study the natural language and the way machines manipulate it, a simply approach when analyzing a text document could rely on frequency statistics such as the Term Frequency-Inverse Document Frequency (TF-IDF) [24]. From a representation learning prospective, text can be considered as a 1-dimensional (1D) grid where elements have a sequential dependencies, and similarly as a time series, this representation cen be analyzed through the use of DL models such as Recurrent Neural Networks (RNN) architecture [126].

Computer Vision (CV), that study how to acquire, process and analyze digital images, can approach the analysis by studying the distribution and the co-occurrence of the gray-levels of the pixel [122]. A more natural approach can be to consider the images as 2-dimensional (2D) grid where elements have local positional dependencies, allowing the usage of DL models such as the Convolutional Neural Networks (CNN) to perform more fine segmentation [107]. NLP and CV both have benefited from the rise of Transformers [219] that allows to deeply discover the intrinsic dependencies of the considered input grid structures [117, 194].

1.2.2 Representing data with graphs

Graphs are a data structure that offer a systematic way to represent entities (as nodes) and explicitly model their relational structures (as edges), providing them with a different meaning depending on the domain of interest. The terms *graph* and *network* are sometimes used interchangeably, however, a network generally refers to the connection between some entities in real systems, such as a document connected by URL in the World Wide Web (WWW), while graph is used to discuss the mathematical representation of networks [15].

The extraction of structural information to obtain graph-related features has historically been associated with graph statistics from the field of graph theory. These measures are primarily used to characterize the role and the importance of the graph elements [79, 85], to find similarity between them according to their proximity in the graph [135, 84], and to identify substructures of interest from the graph itself [2, 139].

The introduction of new graph-centered computational models has revolutionized the way graph structures are processed in ML applications, introducing the field of graph representation learning (GRL) [87]. GRL makes the same consideration made by RL model applied on grid-structured data, that are interested in learning a representation from a structure in which the domain is fixed, and extend the paradigm to graph structures, where both the domain and the signals, defined on it, e.g. node attributes, play a crucial role [27].

GRL paradigms have achieved remarkable results in biology and medicine [121]. Topo-

logical Data Analysis (TDA) aims to extracting relevant information from the underlying topology of data projections, and has been successfully used to infer structural phenotypes from complex biomedical datasets by linking patients affected by the same disease according to their similarities [149, 120]. Knowledge Graph Embedding (KGE) models are suitable for learning the representation of Knowledge Graph (KG), a graph data structure that resulted by the harmonization of different biomedical knowledge repositories [35, 181]. By preserving the semantic similarity encoded in biomedical KG, KGE-based learning technique can be used to combine different biomedical entities and to contextualize data representation using medical knowledge [104]. Graph Neural Networks (GNN) are a type of DL architecture that learn graph representation by combining the structural information and the graph element attributes, mostly optimized according to a supervised task, and employed as a predictive model, such as drug activity [203, 46], and disease classification [173, 167].

1.2.3 The Potential of Graph Representation Learning in Biomedical Domains

GRL has becoming a transformative technique in medical informatics and bioinformatics where data are inherently relational and often structured as networks, such as, protein-protein interaction (PPI) and gene regulatory networks [121]. In these domains, integrating structured biomedical knowledge from application-oriented KGs, such as PrimeKG for precision medicine [35] and ComptoxAI [181] for computational toxicology, or from domain medical knowledge, such as an identified disease phenotype given by medical examination, with real-world data is essential for robust inference and discovery.

In patient risk stratification, GRL data-driven models can be applied on real-world data deriving from clinical studies such as EHR, and augmented with the phenotypic medical definition as a semi-supervised signal, to discover patterns while considering the patients' initial clinical evaluation. When dealing with phenotype classification, GRL models can combine real-world data such as EHR and omics data, and provide a new contextualized patients' representation, by leveraging knowledge from KGs and incorporating prior biomedical understanding between the variables. Similarly, when considering predicting drug activity, GRL enables to embed molecular structure, atom and edge attributes, and to combine them with known molecular pathways and high-throughput screening data, fusing the domain data space with semantic information space from KGs in a hierarchical way.

By bridging knowledge structural representation with data-driven methods, GRL offers a scalable and flexible framework for tackling complex biomedical tasks dealing with real-world data from clinical variables(**Aim 1**), their combination with omics data (**Aim 2**), and

from publicly available toxicology repository (**Aim 3**), offering to improve model predictive performances and the interpretability of the results, supporting decision-making in healthcare and life sciences [62, 104].

1.3 The INTESTRAT-CAD project

This project has been funded by Fondazione Regionale per la Ricerca Biomedica (FRRB), Research Grant no. CP2 14/2018, PI: Gualtiero I. Colombo. The INTEgrated STRAtification Tools in Coronary Artery Disease (INTESTRAT-CAD) project involves different Italian partner, comprising the University of Pavia, and guided by the Monzino Heart Center, in Milan. The project has the goal to integrate different tools for the stratification of the CAD, with main hypothesis that the combination of biomarker identification and the use of imaging analysis, make it possible to monitor the development of atherosclerotic lesions, potentially prevent fatal outcomes.

1.3.1 Coronary Artery Disease

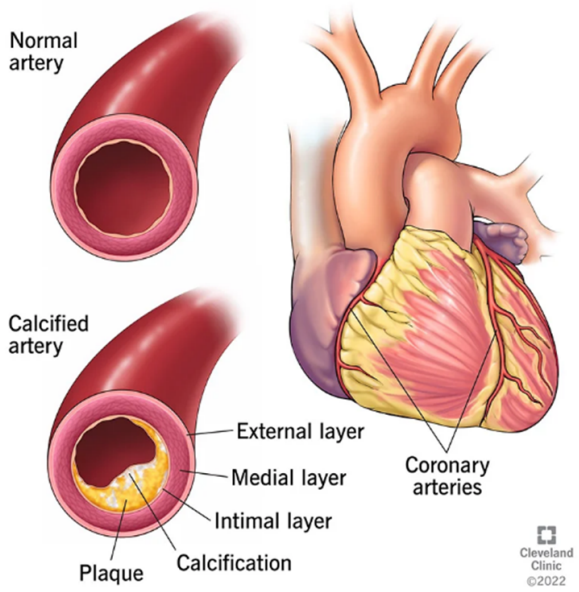


Figure 1.1: Coronary arteries are the vessels that supply oxygen and nutrients to the myocardium (right); the calcification accumulated in the coronary lumen leads to a formation of a atheromatous plaque (left) [45].

Coronary Artery Disease (CAD) are various pathological conditions that affect the coronary arteries. The main cause of CAD is atherosclerosis, involving the formation of athero-

matous plaques in the vessels. The diagnostic management of suspected CAD starts with the assessment of symptoms and traditional risk factors, continues with the execution of basic examinations, e.g. electrocardiogram, stress test, and biochemical tests, and finally estimates the clinical likelihood of obstructive CAD, i.e. high-grade coronary lumen stenosis, to select patients who could benefit from myocardial revascularization [223].

Coronary Computed Tomography Angiography (CCTA) constitutes the state-of-the-art (SOTA) tool for the noninvasive evaluation of CAD [138], enabling the early quantification of coronary artery stenosis (CAS) [145], showing remarkable prognostic values in asymptomatic patients [208]. However, coronary lesions that are not hemodynamically significant may be responsible for acute coronary syndromes, and it became necessary to understand which factors contribute to triggering the acute event or vice versa [140].

Beyond the single coronary plaque, the assessment of the global atherosclerosis burden has been suggested to have prognostic value. Current algorithms to estimate the pretest probability of CAD are largely based on a set of well-defined clinical risk factors [156, 193]. However, traditional risk scores show sub-optimal predictive capacities, and the accurate identification of "at-risk" individuals remains a major challenge [77]. The identification of biomarkers, e.g. from RNA-sequencing (RNA-seq), gives a prominent objective for developing strategies for CAD diagnosis since these are related to the injured tissue [146]. Specifically, peripheral blood gene expression profiling is an informative approach to investigate disease-specific states, study inflammatory responses, and identify biomarkers that reflect disease severity and activity [103].

1.3.2 The INTESTRAT-CAD study design

The INTESTRAT-CAD project is based on an ongoing prospective, observational study, called [13], aimed at predicting the first clinical manifestation of non-obstructive coronary atherosclerosis in an individual. The cohort involved in this trial comprise more than 1,000 adult patients with suspected CAD and without history of coronary events, or revascularization, enrolled at the Monzino Heart Center between the 2016 and the 2021.

The INTESTRAT-CAD project's Population-Investigation-Comparison-Outcomes (PICO) framework, generally used to describe evidence-base medical studies [176], is the following:

- **P**: patients enrolled in Epifania, comprising subjects with non-obstructive (at least one stenosis between 25-50%), subclinical (at least one stenosis between 50 and 70% clinically silent or with inconclusive provocative myocardial ischemia tests), and obstructive coronary CAD (at least one stenosis more than 70%);
- **I**: the initial CAD phenotype is assessed during a first visit with SOTA diagnostic tool,

then this evaluation is repeated in a second follow-up visit a 2-3 years to investigate the progression of the CAD;

- **C**: the Epifania's cohort comprises subjects without coronary atherosclerosis that are used as negative controls, in addition obstructive CAD patients are used as positive controls;

- **O**: the INTESTRAT-CAD project has the aims of identify sub-classes of patients within CAD, integrate data from different source such as clinical examination, multi-omics and imaging, to achieve a better and more complete characterization of CAD patients, and develop technologies for the analysis and subsequent validation of the huge amount of data generated by the study, consisting in a unique bio-bank of CAD patients undergoing deep phenotyping and genotyping.

During the enrolment visit, CCTA is used to assess the clinical suspicion of CAD for each patient, and the value of the most clinically relevant CAS is computed, identifying the patient's CAD phenotype. Baseline clinical information, comprising demographics, medical history, questionnaires assessing diet habits, social support and depressive status, and blood laboratory measurements, is collected and saved into the REDCap Case Report Form (CRF). RNA-sequencing (RNA-seq) from a peripheral venous blood sample is performed to obtain gene expression data.

1.4 Computational Toxicology for small molecules

1.4.1 Small molecules as potential drugs candidates

Small molecules are organic compound with a low molecular weight $\leq$ 1000 Daltons (Da) and a size on the order of 1 nm. Giving their low weight structure, these molecules can be rapidly diffuse across cell membrane to reach intracellular sites of action, making them suitable drug candidates that can be tailored to achieve specific therapeutic targets [200]. About 90% of all drugs sold are small molecules, which, thanks to their low weight, are cheaper and easier to synthesize than biologic drugs and can be administered orally [128].

However, even if small molecules are easy to synthesize, traditional drug discovery and development is known to be time consuming and cost-intensive, with a reported low rate of success of those drugs undergoing clinical trials [57]. Evaluate possible drug adverse effects in an early stage of the drug development process became important to increase the success rate as well as reduce time in screening candidates [189].

1.4.2 Tox21 consortium

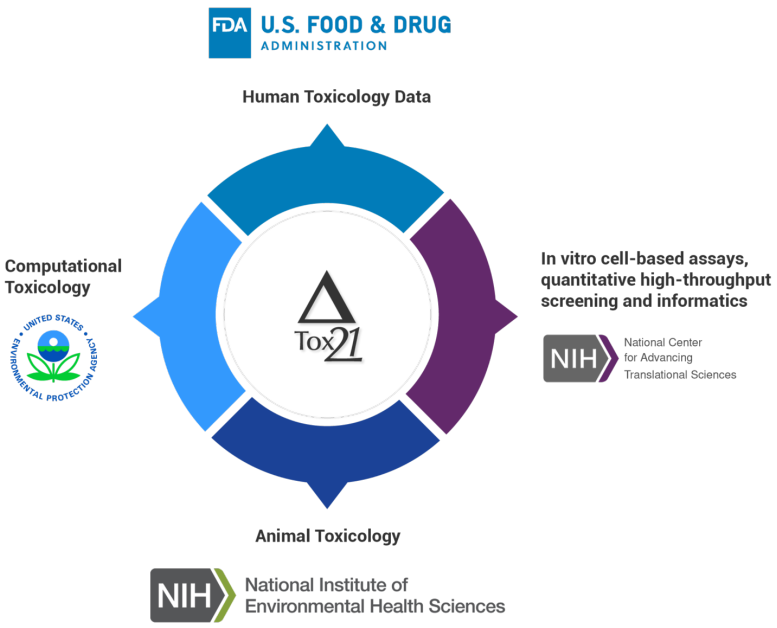


Figure 1.2: Toxicology in the 21^{st} century (Tox21) is a U.S. consortium with partners that bring different expertise, including animal toxicology from the National Toxicology Program (NTP) at the National Institute of Environmental Health Sciences, Computational toxicology from the Environmental Protection Agency (EPA), human toxicity data from the Food and Drug Administration (FDA), and in vitro cell-based assays, quantitative high-throughput screening and informatics (qHTS) from the National Center for Advancing Translational Sciences (NCATS) [214].

Toxicology in the 21^{st} Century (Tox21) is a U.S. federal research consortium formed in 2009, with the general objective of developing method to rapidly and efficiently evaluate the safety of commercial chemicals, pesticides, food additives, contaminants, and medical products [211]. This consortium allows the construction a of compound library that today contains approximately n = 9,000 tested chemicals, which adverse effects is evaluated with more than n = 80 quantitative high-throughput screening (qHTC) in vitro assay [174].

Each assay has a biological molecule or pathway to monitor or perturb, that defines its target and the assay type, and is performed on a specific cultured cell that reflect different tissue or organ systems of interest. For example, cytotoxicity assays measure the ability of compounds to cause cell damage or cell death. Apoptosis is a process of programmed cell death, and plays a central role in the development of multicellular organisms; this is a highly regulated process and induces cell death by activating members of cysteine aspartic acid-specific protease (caspase) family of enzymes. Caspases involved in apoptosis are classified

by their mechanism of action as initiator caspases and executioner caspases. Inhibition of apoptosis results in number of cancers, autoimmune diseases, inflammatory diseases and viral infection. The *tox21-casp3-cho-p1* assay is used to identified small molecules that induce apoptosis, where wild type CHO-K1 hamster cells are used to scree the Tox21 libraries for identifying caspase-3/7 activity inducers [212].

The data are publicly available [213], making more accessible the exchange of qHTS activities, that are traditionally considered as the primary domain of the pharmaceutical industry.

1.4.3 Evaluate the chemicals toxicologic effects with computational methods

Assessing the toxicologic effects of chemical substances can be expensive and time consuming considering the vast number of chemicals of toxicological concern [165].The field of computational toxicology has emerged, giving new approaches to evaluate chemical toxicology in silico [172].

The most widely used method to evaluate toxicity of chemicals in silico is Quantitative Structure-Activity Relationship (QSAR) modeling [59], where quantitative descriptors of molecules' structures are used as input to train a computational model in predicting toxic endpoints of interest. The descriptors that are typically used include simple molecule properties, such as number of rings and molecular weight, molecular fingerprints[141, 60, 178] (i.e., bitstrings that indicate the presence of particular molecular substructures or fragments), or high-dimensional vector representations, called embeddings, obtained using DL [233] or similar techniques. Considering this kind of input, QSAR is often criticized to not capturing relevant structural features of chemicals and model interpretability, demanding the need for new methodological innovations [42].

Most current research including toxicity predictions adopt exclusively data-driven methods, thus neglecting the wealth of external semantic knowledge resources that have the potential to improve prediction performance and interpretability of the results [180]. These resource are informatics KBs containing the scientific evidence related to known toxicity outcomes, such as the Comparative Toxicogenomics Database (CTD) [53] and Adverse Outcome Pathway Database (AOP-DB) [143], and suitable frameworks that combine them such as ComptoxAI [181].

1.5 Objectives and aims of the thesis

The objective of the thesis is to propose AI frameworks based on GRL methods and integrating pre-existing knowledge, to address the analysis of biomedical data from the INTESTRATCAD projects and computational toxicology. The main motivation behind the choice of GRL is that it provides methods that balance between the flexibility of data-driven methods, such as clustering algorithm and predictive models, and the structured data representation, that is an advantage for the integration of multiple data types [62] and the medical knowledge [104].

Aims are reported in Figure 1.3, where for each of them are reported the type of data used, the kind of knowledge integrated in the analysis, the method employed to implement the aim in blue, and the task performed in red.

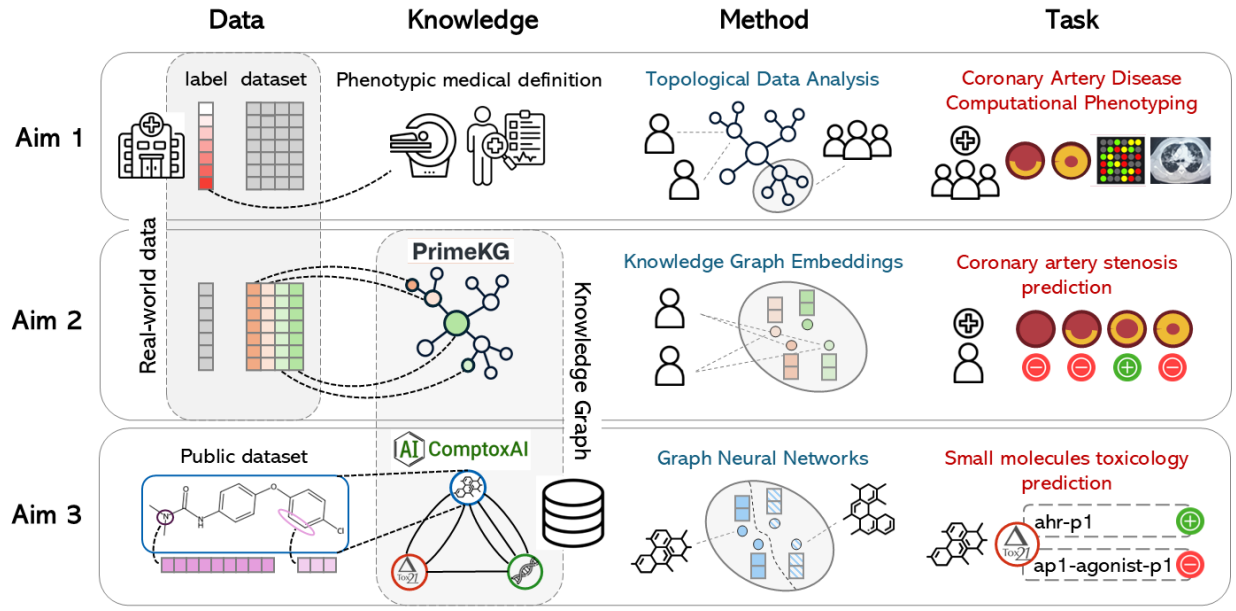


Figure 1.3: Aim 1 and Aim 2 rely on real-world data from the INTESTRAT-CAD project, which includes a patient dataset with labelled CAD phenotypes. Aim 1 applies a semi-supervised TDA framework guided by the medical knowledge of the initial CAD phenotypic definition to perform computational phenotyping. Aim 2 and Aim 3 both integrate KGs; Aim 2 employs PrimeKG and KGE model to contextualize with pre-existing medical knowledge the CAD patients' representation, therefore used to predict coronary artery stenosis, while Aim 3 uses public chemical compounds data enriched with ComptoxAI KG and applies GNNs for molecular embedding and toxicity prediction.

Three different applications are reported, each addressing a specific biomedical task, from stratification and computational phenotyping (**Aim 1**) to prediction of coronary artery stenosis for risk stratification (**Aim 2**), and small molecules toxicology prediction (**Aim 3**).

In particular, **Aim 1** and **Aim 2** use both real-world data from the INTESTRAT-CAD project, defined as the patients' dataset variables and the initial CAD phenotype assigned to the CCTA images, while **Aim 3** uses Tox21 data that is a public repository on compounds data.

However, the initial phenotypic medical definition is used as pre-existing knowledge in a semi-supervised framework for computational phenotyping (**Aim 1**), and true label to train a predictive model in coronary artery stenosis prediction (**Aim 2**). In addition, **Aim 2** and **Aim 3** share the same source of biomedical knowledge, by leveraging KGs. **Aim 2** uses a precision medicine-oriented KG, i.e., PrimeKG, to contextualize the patients' clinical and omics variables with medical knowledge, while **Aim 3** adopt ComptoxAI, to augment the chemical data with semantic relation between molecules, toxicological assay and genes.

Identifying these set of data and knowledge bases to adress the tasks, these thesis reaches its objective adopting specific GRL paradigm for each one of the Aims, from Topological data representation (**Aim 1**) to graph embedding, specifically shallow method suitable for knowledge graph (**Aim 2**) and neural networks-based models (**Aim 3**). The integration of pre-existing medical knowledge is formulated according to the application considered:

- **Aim 1.To perform CAD computational phenotyping on the basis of clinical studies data and domain medical knowledge with TDA:** considering real-world dataset, the label expressing an initial phenotypic medical definition can be used to adapt data-driven method GRL such as TDA, in a semi-supervised fashion. The idea is proposed in a tool called *pheTDA*, presented during the 21st International Conference of Artificial Intelligence (AIME) 2023 [7]. pheTDA is a semi-supervised TDA-based framework to assist the computational definition of novel phenotypes. Considering a patients populations affected by the same disease, pheTDA guides the hyperparameter tuning of the TDA Mapper pipeline by evaluating the graph entropy, measured by averaging the node entropy that measure the presence of patient initial phenotypes in each node. The tool identifies subgroups of patients from the topology by using community detection methods, and assess discriminative features for each subgroup of patients via predictive models. pheTDA is applied on a subset of the initial INTESTRAT-CAD cohort, by considering their clinical variables, and identifying five novel subgroups, one of them only comprised diabetics patients with a higher CV.

- **Aim 2.To develop a predictive risk model of CAD combining omics and clinical data with a precision medicine oriented KG and KGE:** the accurate diagnosis of CAD should ideally combine multimodal information from clinical, laboratory, and omics-based data streams. GRL methods are particularly promising,

integrating data-driven approaches with prior biomedical knowledge. In particular, a work that will be presented during the 23st International Conference of Artificial Intelligence (AIME) 2025, investigates the use of PrimeKG, a KG proposed for precision medicine, to address the classification of CAS. Specifically, clinical variables and whole blood transcriptome profiles, belonging to the INTESTRAT-CAD cohort are mapped to the PrimeKG entities, while KGE models are adopted to learn graph representation. This strategy acts as a fusion strategy for the patients variables, considering representing each patients as a combination of the entity embeddings corresponding to the value of his/her feature in the dataset. The application show that this knowledge-enriched patient representation is useful in the classification of CAD severity when used as input to predictive models, by improving the classification performance when compared to classic data-driven strategies in both single and multi-modal settings;

- **Aim 3.To augment a toxicity prediction model for small molecules combining semantic knowledge between chemicals, gene and Tox21 assays with a computational toxicology KG and GNN:** considering recent advances in publicly available computational toxicology knowledge resources, the object of this study, that is currently under revision, is to investigate the effect of adding semantic data to QSAR modeling for toxicity prediction. ComptoxAI is one recently published KG for computational toxicology, and it is used to first build a dataset of 2D molecular structures, and second to create a heterogeneous graph with chemicals, Tox21 assay and genes nodes, that are connected with semantic relations. GNN are used to: i) learn the molecules embedding, by leveraging its graph structure, ii) propagate the semantic knowledge in the graph, iii) to make the small molecules toxicology predictions for the selected assay, and iv) infer the most informative chemical substructures pertaining to the assay of interest. The predictive performance of these strategies are compared with baseline ML models and GNN ones that do not use the semantic knowledge, showing that the proposed strategy improve both discrimination and calibration. Lastly, the application shows some examples of explanations obtained with a graph-based XAI method, identifying the most important subgraph involved in the prediction task.

1.5.1 Thesis structure

In Chapter 2, it is given first a definition of graphs, some types of them according to specific characteristics, by reporting some examples in Section 2.1, and the main ML tasks formulated on these data structures in Section 2.1.1. Then, the GRL methods adopted in this these are introduced, starting from traditional methods based on statistics, i.e. graph the-

oretic techniques (Section 2.2), non-linear graph-based dimensionality reduction techniques, i.e. Manifold Learning methods (Section 2.3), Topological Data Analysis (Section 2.4) and methods based on Neural Networks (NN) for learning the graph embeddings, comprising both Shallow methods (Section 2.5.1) and Graph Neural Networks (Section 2.5.2), with an additional focus on Knowledge Graph Embedding methods (Section 2.5.3). For each of the method introduced, it is also reported a section containing a literature review with some biomedical application of interest.

Following, each application is described in detail in a dedicated chapter, composed by a brief *Introduction* section, a *Material and Methods* section in which the study framework, the data used and the method implemented are described, a *Results* section reporting the different results of the study, and a *Discussion* section in which the findings of the study, the limitations and the future works are reported and discussed. These chapters are: Chapter 3 describe how to perform CAD computational phenotyping with a TDA-framework, by considering as input data the clinical variables, and the CAD phenotypic definition as pre-existing medical knowledge (**Aim 1**), Chapter 4 deals with the develop of a CAS prediction model that combine clinical and omics patients' data through the use of KGE models (**Aim 2**), and Chapter 5 describes the augmentation of a toxicology predictive model perfomance and interpretability of its results, with the use of GNN to both incorporate semantic knowledge and to learn small molecules (**Aim 3**).

Lastly, in Chapter 6, the conclusions are reported, by highlight the motivations that lead to the choice of adopting GRL methods to build AI models that combine data-driven with structured knowledge 6.1, then summarizing the main findings in Section 6.2, and finally by discussing some possible future developments of the AI framework developed, by considering both applicative opportunities and methodological improvements in Section 6.3.

Chapter 2

Methodological Background

2.1 Graph definition and notation

A graph is a data structure defined as $G = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ constitutes the set of the vertices, or nodes, with $|\mathcal{V}| = N$ defining the number of the nodes, and $\mathcal{E} = \{e_1, e_2, ..., e_M\}$ is the set of the edges, or relations, with $|\mathcal{E}| = M$ defining the number of the edges. The term *subgraph* refers to a subset $S = (\mathcal{V}_S, \mathcal{E}_S)$ of the graph G, where $\mathcal{V}_S \subseteq \mathcal{V}$ and $\mathcal{E}_S \subseteq \mathcal{E}$. In addition, a sequence of nodes with a specific length l and connecting two nodes v_i and v_j, indicated as $(v_i \to ... \to v_j)_l$, is called a *walk*; specifically the sequence is defined a *path* when made by distinct nodes.

Given a node v_i, its *neighbor* is defined as the collections of nodes connected to it, while the *k-hop neighbor* is the set of nodes that distance exactly k connection or *hops* from the node, defined as:

$$N_{v_i}^k = \{v_j | d(v_i, v_j) = k\}, \tag{2.1}$$

where $d(v_i, v_j)$ is the distance between two nodes v_i and v_j.

A graph G can be represented by its *adjacency* matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, in which each element $a_{ij} \neq 0$ if an edge exists between the nodes v_i and v_j. It is also convenient to define the *Laplacian* matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$, in which each element l_{ij} is defined as:

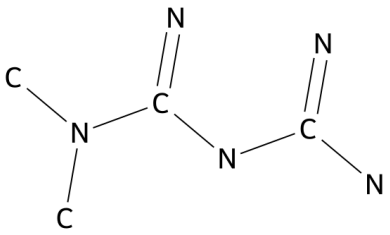
$$l_{ij} = \begin{cases} k_{v_i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}, \tag{2.2}$$

with k_{v_i} represent the *degree* of the node v_i, i.e. the number of nodes that have a direct connection to it. Note that $\mathbf{L} = \mathbf{K} - \mathbf{A}$, where $\mathbf{K} \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix,

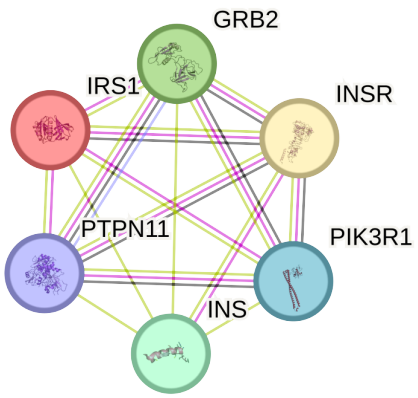
where each element represent the degree of a node in the graph.

Depending on specific properties, a graph can be defined:

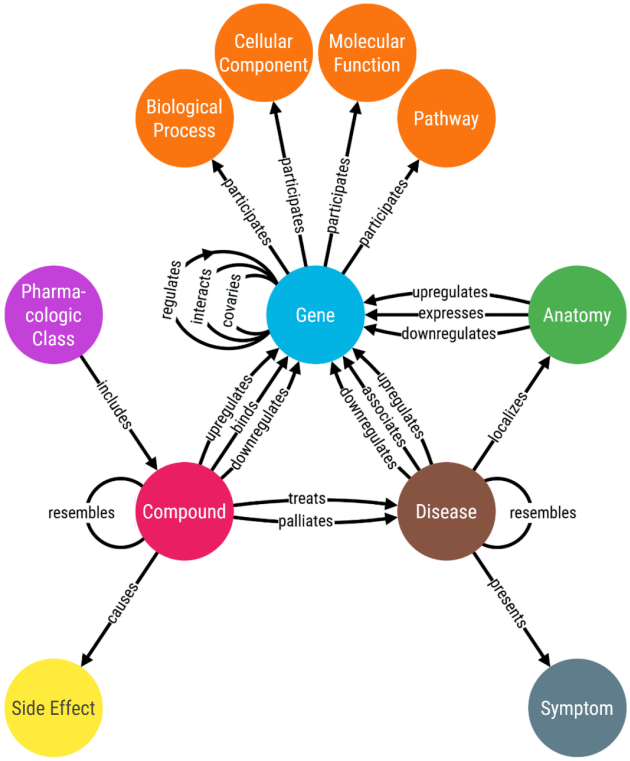
- *directed*: if the edge have a direction, otherwise the graph is defined *undirected*;
- *weighted*: depending on whether the set of edges is associated with single-dimensional weights, used to emphasize specific connections. Otherwise a graph is defined *simple*;
- *attributed*: if the nodes and/or the edges hold single- or multi-dimensional attribute, or features, describing node or edge properties. In case the graph holds node features with dimension F, these are represented by a matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$, while edge features with dimension $F_{\mathcal{E}}$ are represented by a matrix $\mathbf{E} \in \mathbb{R}^{M \times F_{\mathcal{E}}}$;



(a) Molecular structure of Metformin with the hydrogen atoms omitted, obtained with the RDKit Python package[170].



(b) PPI network extracted by querying the STRING DB with Insulin receptor substrate 1 (IRS-1) as input [207].



(c) The *metagraph*, or graph schema of the Hetionet KG, having as nodes the semantic types of the KG nodes and the name of the relations between them with an arrow between nodes [89].

Figure 2.1: Examples of graphs in the biomedical field: molecular structure of Metformin is an example of ndirected, homogeneous, and attributed graph (a); a small PPI network extracted from the STRING DB can be viewed as an undirected, heterogeneous and weighted graph (b); the metagraph of Hetionet KG, an heterogeneous graph containing biomedical knowledge (c).

- *heterogeneous*: this adjective is used to define graph in which the nodes and/or the edges belong to different types. In this case, the set of the node types is defined by $\mathcal{A}$ while the set of relation types is defined by $\mathcal{R}$. A special case is when the graph comprises two types of nodes and edges connect node from different type, defining a *bipartite* graph. In case of nodes and edges with the same type the graph is defined *homogeneous*;

- *knowledge graph* (KG): this is a kind of heterogeneous graph which definition recall the atomic data entity of the Resource-Description framework (RDF) [131], and consider the graph as made by a set triples of the form $(v_i, r, v_j) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, in which *head* entity v_i and the *tail* entity v_j are connected with a relation of the type $r \in \mathcal{R}$.

As an example, a molecular structure can be intuitively described as an undirected, homogeneous, and attributed graph (Figure 2.1a), in which the nodes and the edges, atoms and bonds, respectively, belong to the same semantic types, but characterized with different attributes that specify their structural types (e.g. atomic number, type of bond). Differently, the protein-protein interaction (PPI) network reported in Figure 2.1b, can be view as an undirected, heterogeneous, and weighted graph, in which the proteins are connected with different types of interactions, resulting from different empirical evidence, reported with different edge colours, and weighted with the degree of scientific evidence. Lastly, Figure 2.1c reports an example of KG, used in the biomedical field to capture the knowledge from literature and biomedical repositories. In this kind of graph structure, the nodes represent biomedical entities, these are connected with directed relations that represent functional interactions between them, such as Compound *binds* Gene and Gene *participates* in Pathway.

2.1.1 Machine learning task on graph structure

Given a graph $G = (\mathcal{V}, \mathcal{E})$, it is possible to define some ML task, that can be divided in four broad categories: graph prediction, subgraph detection, latent graph learning and graph generation:

- *graph prediction* aims at predicting a true label that can be defined on the nodes, edges, to the graph itself or to a subgraph from it. Specifically, for each of them it is possible to learn a function $f : * \longrightarrow \mathcal{Y}$, where the symbol $*$ indicates the input space, and $\mathcal{Y}$ the set of labels; toghether, these elements define the prediction task, that can be a classification or a regression, depending if the label to predict is categorical or numerical. For example, in node classification the learned function f maps the set of the nodes to a set of categorical label, while in graph regression f maps a set of graphs

to a set of numerical label. In addition, it is possible to define the link prediction task, where the function to learn is of the form $f : \mathcal{E} \cup \mathcal{E}^C \longrightarrow \{0, 1\}$, and the task consists of predicting the existence of a link between two nodes. Note that all the graph prediction task previously mentioned involves a label and are defined *supervised* in ML.

- *subgraph detection* is the task in which a set of substructures $\mathcal{S} = \{S_1, S_2, ..., S_s\}$ are identified in the graph. Generally, the substructures identified are called *modules*, when they contribute to a specific variable or if their union does not constitute the entire graph but a part of it, meaning that $\{S_1 \cup S_2 \cup ... \cup S_g\} \subseteq G$. Differently, the subgraph identified are referred as *communities* when each of them contains similar nodes or if their union make the overall graph, that is $G = \{S_1 \cup S_2 \cup ... \cup S_g\}$. The task of identifying communities is called community detection but is also referred as graph clustering, for which is not defined an a-priori ground truth partition, i.e. the label task, and for that this task is defined *unsupervised* in ML.

- *latent graph learning* learn a function $f : \mathcal{V} \longrightarrow \mathcal{E}$ that map the set of the nodes with the edges, meaning that learn the underlying graph structure. The learned graph can be used to perform prediction task, obtain the data topology, and generates low dimensional representation of the feature attributes. The task of latent graph learning can be formulated both as a supervised task, if the edges are given, or unsupervised task, if they are not.

- *graph generation* considers the objective of generating new graph structures with some properties of interest. In particular, given $\mathcal{G}$ a set of training graphs $\{G_1, G_2, ..., G_g\}$, the task is to learn a function $f : \mathcal{G} \longrightarrow \mathcal{D}_G$ that obtains a distribution $\mathcal{D}_G$, characterizing the training graphs, that can be used to generate new graphs $\tilde{G}$. Being a generative task, this is an unsupervised task for its nature.

Note that for each of the introduced task, the function f learned is reported for simplicity as acting act on graph structural units, such as node and edges, but can also include their feature matrix as inputs.

2.2 Graph theoretic techniques

The study of graphs is referred as the field of graph theory or network science [15]. Traditionally, the role of the nodes, the edges and the different graph substructures, can be assessed by using graph theoretic techniques, that are functions that map the graph component to

real values, representing different aspects of the graph. These measures can be categorized depending which graph element is considered, and are described in the following sections.

2.2.1 Node-level

The roles of the nodes in a graph can be generally addressed by evaluating its *centrality*, that is a concept that relates the importance of a node to its position in the graph. Node centrality can be defined according to different criteria, as showed in Figure 2.2, leading to different centrality measures. The most popular ones are reported in Table 2.1.

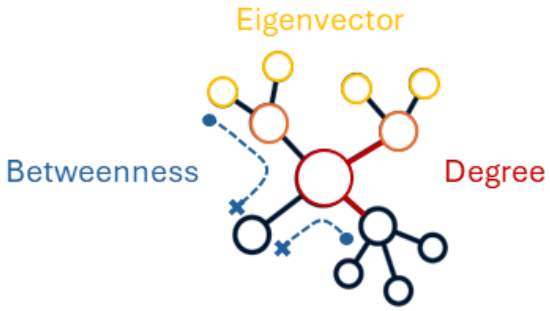


Figure 2.2: Statistics to evaluate node centrality are based on different criteria, such as the direct connection with adjacent nodes (Degree centrality represented in red), the number of times it appears in a short path (Betweenness centrality in blue), or by looking at high-order connection than the first (Eigenvector centrality indicated by the change of colour from red to yellow).

An intuitive way to consider the centrality is by counting the number of connections a node has, defining the node *degree*, reported in Eq. 2.4. Differently, *betweenness* centrality computes the centrality of a node using the concept of shortest path, indicated as:

$$min(v_i \rightarrow ... \rightarrow v_j), \tag{2.3}$$

and defined as the minimum sequence of nodes (and edges) that connect two given nodes. In particular, the *betweenness* for a node v_i, reported in Eq.2.5, compute $\#min(v_j \rightarrow v_i \rightarrow v_q)$, that is the the number of shortest paths between the nodes v_j and v_q that pass through v_i, out of $\#min(v_j \rightarrow ... \rightarrow v_q)$, that is the total number of shortest paths between the nodes v_j and v_q. The nodes with high degree are usually called *hubs*, and identify centers of connections in the graph, while the nodes with high betweenness are called *bottlenecks* since controls the flow of information in the graph.

A different category of measures is based on the eigenvector and eigenvalues transformation of the matrix underlying the graph structure. For example, for the adjacency matrix $\mathbf{A}$

the transformation is $\mathbf{At} = \lambda\mathbf{t}$, from which the centrality for node v_i can be defined as the -th element of a left eigenvector $\mathbf{t}$ associated with the largest eigenvalue λ of $\mathbf{A}$. This is the main idea behind Eigenvector centrality, that is originally formulated for undirected graphs but can be extended in the directed case, that simply computes the centrality of a node v_i by computing the centrality of the nodes connected to its beyond the first connection, while applying an iterative procedure to resolve the Eq. 2.6.

Centrality measure name	**Equation**	
Degree	$k_i = \sum_{j=1}^{N} a_{ij}$	(2.4)
Betweenness	$BC_i = \sum_{i \neq j \neq q} \frac{\#min(v_j \to v_i \to v_q)}{\#min(v_j \to \ldots \to v_q)}$	(2.5)
Eigenvector	$\mathbf{t}_i = \frac{1}{\lambda} \sum_{j \in V} a_{ij} \mathbf{t}_j$	(2.6)
Laplacian	$CL_i = \frac{E_L(G) - E_L(G_i)}{E_L(G)}$	(2.7)
PageRank	$PR_i = \gamma \sum_{j \in V} \frac{a_{ij}}{k_j^{out}} PR_j + \frac{1-\gamma}{N}$	(2.8)

Table 2.1: Principal centrality measures used to characterize the role of the nodes in a graph.

PageRank centrality (Eq. 2.8) is created to rank web pages in Google retrieval [155], and is a modification of the Eigenvector centrality suitable for directed graphs, in which the contribution of each node v_j is weighted by its out-degree k_j^{out}, that is the number of edges that originate from the node, and the dumping factor $\gamma \in [0, 1)$ is introduced. Laplacian centrality (Eq. 2.7) is proposed for weighted graphs, and it is based on the drop of the Laplacian energy $E_L(G)$ resulting after deleting the node v_i from the graph G, with $E_L(G) = \sum_{i=0}^{n} \lambda_i^2$ and λ_i the i-th eigenvalues of $\mathbf{L}$ [164].

Biomedical application of centrality measures

Centrality measure can be used to characterize the roles of the nodes in biomedical graph. For example, when dealing with comorbidity networks, where nodes represent pathological condition shared between patients, through centrality measures it is possible to identify the most prevalent ones characterizing the population in analysis [85].

Starting from a knowledge repository, such as the Online Mendelian Inheritance in Man (OMIM) [11], that collects literature and evidence about disorder and genes, it is first possible to create a bipartite networks between these entities, and then use centrality measure to identify hub disease, expressing common phenotypic condition between several disorders [79]. When considering PPI networks, centrality measure are often used to identify hub nodes, however, the role of bottleneck has been suggested to be essential, since act as a key connection for functional and dynamic properties between the genes [241].

While centrality measures showed to be positively correlated for a broad range of networks from different domains [152], some of them found specific fields of applications. PageRank centrality is widely adopted in the analysis of biological networks [101], for example can be used in bioinformatics pipeline for the identification of gene biomarkers, involved in latent regulatory changes and that are in common between different diseases [12]. Eigenvector centrality is widely adopted when dealing with networks in which an higher number of high degree nodes is connected to lower degree nodes, or vice-versa, and can be used when working with networks expressing functional brain regions [186]. However, since there is no general rule when choosing a measure, in practice one could employ several of them to identify nodes with different functionality and then assess the overall node centrality [225].

2.2.2 Edge-level

Edge-level statistics can be derived from node centrality measure; however, it could be more interesting focus on statistics that evaluate the *proximity*, a concept that is close to the meaning of the distance between two nodes in a graph. The most popular metrics used to assess nodes proximity are reported in Table 2.2.

Figure 2.3: Nodes proximity is evaluated at a edge-level by using statistics that are based on the geodesic distance between two nodes (in blue), that is the length of the shortest path connecting them, or by assessing the similarity according to their neighboring nodes (the two red nodes interstate in their 2-hop neighbours in the purple node).

Traditional statistics, consider the neighbors of the nodes to assess the proximity. Note that for simplicity a 1-hop neighbor is considered in the measures reported in Table 2.2, omitting the k from the Eq. 2.1. The Jaccard coefficient reported in Eq. 2.9, is a popular measure used to evaluate the similarity between sets, and can be used to evaluate the proximity of two nodes v_i and v_j, computed as the ration between the size of the intersection between neighbors N_{v_i} and N_{v_j} and their union. The Overlap coefficient (Eq. 2.10) only considers the intersection between the neighbors, diving it for the size of the smallest neighbors

between the two. Similarly, the Adamic-Adar index [1] reported in Eq. 2.11, is computed as the sum of the inverse logarithmic degree centrality for each node v_q that belong to the intersection of the two neighbors.

Name	Equation	
Jaccard coefficient	$Jacc_{ij} = \frac{\lvert N_{v_i} \cap N_{v_j} \rvert}{\lvert N_{v_i} \cup N_{v_j} \rvert}$	(2.9)
Overlap coefficient	$OC_{ij} = \frac{\lvert N_{v_i} \cap N_{v_j} \rvert}{min(\lvert N_{v_i} \rvert, \lvert N_{v_j} \rvert)\rvert)}$	(2.10)
Adamic-Adar index	$AA_{ij} = \sum_{q \in N_{v_i} \cap N_{v_j}} \frac{1}{log\lvert k_q \rvert}$	(2.11)
Closest	$d(\mathcal{V}', \mathcal{V}'') = \frac{1}{\lvert \mathcal{V}'' \rvert} \sum_{v_j'' \in \mathcal{V}''} min_{v_i' \in \mathcal{V}'} \lvert \min(v_i' \to ... \to v_j'') \rvert$	(2.12)
Shortest	$d(\mathcal{V}', \mathcal{V}'') = \frac{1}{\lvert \mathcal{V}'' \rvert} \sum_{v_j'' \in \mathcal{V}''} \frac{1}{\lvert \mathcal{V}' \rvert} \sum_{v_i' \in \mathcal{V}'} \lvert \min(v_i' \to ... \to v_j'') \rvert$	(2.13)

Table 2.2: Principal measures used to characterize the proximity of a pair of nodes in a graph.

Alternatively, the proximity between two nodes can be computed in a more direct way by considering the geodesic distance between a pair of nodes, defined as the length of the shortest path (Eq. 2.3), and indicate as:

$$\lvert min(v_i \to ... \to v_j) \rvert. \tag{2.14}$$

From these consideration, it is possible to build different measures, suitable to asses the proximity of two different subsets of nodes $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{V}'' \subseteq \mathcal{V}$. For example in Eq. 2.12 is reported the closest measure, defined as the average length of the shortest paths connecting each node in $\mathcal{V}'$ with its nearest node in $\mathcal{V}''$, while in Eq. 2.13 is reported the shortest measure, representing the average length of the shortest paths connecting each node in $\mathcal{V}'$ with each node in $\mathcal{V}''$ [84].

Biomedical application of node proximity measure

A relevant example of use of proximity measures in the biomedical field is the one related to the interactome based approach to study of human disease. This approach considers creating a large interactome network by combining several PPI networks, and then to identify entities of interest by leveraging known molecular mechanism that are extracted from specialized biomedical repositories [29].

For example, disease-gene annotations extracted from OMIM, can be used to relate diseases to the set of gene nodes, and measuring the diseases neighbors overlap allows one to identify and describe shared and similar pathological mechanism between them [135]. In

addition, repository containing drug-drug interactions, e.g. DrugBank [113], can be used to identify drugs entities in the interactome network, and proximity measures between set of drugs can be used to make hypothesis about possible new therapeutical target shared between them [41].

Proximity measures are recognized as a well-established approach when dealing with the task of link-prediction, for example in drug-repurposing, i.e. find potential new therapeutical indication of already approved drug. For example, shortest distance can be used as a measure to link drug and disease in a newly proposed KG, to assess its potential capabilities to drug repurposing [35]. With a different approach but with the same purpose, proximity measures can be used to extract features that are subsequently used as input to ML classifier models trained for link-prediction task, for example to obtain Alzheimer's Disease (AD) possible new treatment, that are actually under investigation with already approved clinical trials [182]. However, when compared with more complex approach based on graph-centric predictive models, the use of proximity measure in drug repurposing has shown lower performance [142].

2.2.3 Subgraph-level

Figure 2.4: Considering to identify subgraphs in a network, it is possible to differentiate the identification of motifs, that are patters that recurr in the graph (e.g., the subgraph reported in green), with the task of community detection, in which graph is partitioned according to specific algorithm (in grey).

The characterization at a subgraph-level starts with the identification of specific sub-structures of interest in the graph. A subgraph that have specific characteristics and recur with a pattern in the graph, is defined *motif*. These are considered to be the basic block of more complex biological network. For example, a motif defined by a feedforward loop of three nodes identify transcription in a gene regulatory network [139]. In addition, the

identification of a specific subgraph can be directly suggested by the domain of interest, for example when considering the fragment or functional groups of a molecules [63].

A series of algorithms are proposed in literature for the task of community detections. Generally, these algorithms start by assigning to each node an initial community, then they make a partition or a merging, by deleting or adding an edge, respectively, and a specific metric is evaluated. This process is repeated iteratively until the metric reaches convergence. Community detection algorithm varying primary according to the metric considered. For example, one of the most used method is the Girvan-Newman algorithm [147], which computes, for each edge the betweenness centrality, then removes the edge with the highest value, and repeats the process until the centrality's value is equal for each edge.

A category of community detection algorithms is based on the maximization of a measure called *modularity* [147], that is computed defined for a pairs of node v_i that belongs to a community c_i, and a node v_j that belongs to a community c_j, as the following:

$$Q = \frac{1}{2M} \sum_{ij} [a_{ij} - \gamma \frac{k_i k_j}{2M}] \cdot \delta(c_i, c_j), \tag{2.15}$$

where γ is the resolution parameters, which value is used to weight differently the intra-group and the inter-group edges, and $\delta(c_i, c_j)$ is the Kronecker delta function that evaluates to one if v_i and v_j belongs to the same community, otherwise evaluates to zero.

One of the first proposed algorithm that belong to this category is the Greedy Modularity Optimization method [44], which iteratively merges the pair of communities that yield the highest increase in modularity. This method evaluates the change in modularity for all possible community pairs and greedily selects the one that provides the best improvement. However, this strategy is prone to getting stuck in local optima, often failing to recover the globally optimal community structure.

To address this, the Louvain algorithm [20] introduces a more efficient and scalable approach based on two phases: the local movement phase, where each node is moved to the neighboring community that maximizes modularity gain, and the aggregation phase, where the graph is reduced by collapsing communities into super-nodes. While Louvain improves both speed and the modularity score compared to the greedy approach, it tends to overlook small communities in networks with heterogeneous community sizes, due to its coarse aggregation.

The Leiden algorithm [215] extends Louvain by incorporating an additional refinement phase between the local movement and aggregation steps. In this phase, each community is checked for internal connectivity and, if needed, split into sub-communities before aggregation. This refinement ensures that the resulting partitions are more internally connected

and well-defined, overcoming key limitations of the Louvain method, especially in complex and large-scale networks.

A different family is based on the Label propagation algorithm [47], an algorithm originally proposed as a semi-supervised method to propagate labels from labelled to unlabelled nodes in a graph, where the node label are updated iteratively according to the label of its neighbors [251].

After having identified substructures in a graph, it is possible to characterize their nodes and edges with graph theoretic technique regarding centrality and proximity, as reported in Sections 2.2.1 and 2.2.2, respectively, and then average the values for each subgraph to obtain sub-graph level statistics. Note that this procedure can be generalized to a graph G, allowing to adopt the notation for it. For example, it is possible to compute the graph degree as mean node degree, and according to Eq. 2.4, write it as $k_G = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}$. In addition, other measures can be extracted by summarizing some general property of the graph, such as the density of the graph, expressing the number of connections in a graph and computed as $\frac{2M}{N(N-1)}$, where a value of zero indicates a set of data points without connection and one a complete graph.

Biomedical application at subgraph-level

The identification of recurrent subgraph is of primary interest when dealing with biological networks. For example, the recognition of motifs in PPI networks can be used to learn a signature of specific diseases, and by looking at the adjacent protein nodes is possible to predict the disease-protein association [2]. Without focusing on recurring subgraphs, disease can be broadly represented as modules of node proteins, and compared through the use of proximity measures, computed between the modules, in order to find similar mechanism [135].

Community detection algorithms have found usage in different biomedical fields. Garcia et al. [73] have explored the use of modularity maximization algorithm in brain network analysis by considering functional MRI data. In particular, the authors have focused on how modular structures, identified in brain graphs, correspond to functional and anatomical brain regions, contributing to a better understanding of neural connectivity and cognitive function.

When dealing with bipartite networks, for example one made up of gene-drug interactions, community detection algorithms can be used to reveal clusters of genes and drugs, that showed potential functional or therapeutic relationships [30]. In a different setting, these algorithms are used to identify distinct long COVID sub-phenotypes from Electronic Health Record (EHR) data [52]. By first building a graph where nodes and edges represent patients'

symptoms and their co-occurrence, respectively, the Louvain method was used to detect clusters of nodes with similar temporal patterns, revealing seven distinct data-driven sub-phenotypes of long COVID [52].

2.3 Manifold learning

Manifold learning refers to a collection of methodologies for analysing high-dimensional data that are based on the *manifold hypothesis*, assuming that the data tend to near in a lower-dimensional manifold, defined as a generalization of a curve or a surfaces [67]. Manifold learning methods are used to obtain a non-linear transformation of the data points, that becomes essential when the dataset is high-dimensional, and can be applied for different purposes, such as to facilitate the ML analysis or for simple visualization.

The general pipeline adopted by a manifold learning algorithm is represented in Figure 2.5. The input is a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, in which each sample $\mathbf{x}_i$ is represented by a F-dimensional features. This is also called a *point-cloud*, considered as a set of attributed nodes $\mathcal{V}$ without edges, and for that the same notation given in Section 2.1 for the node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ can be used when referring to it. The objective is to find a mapping $\mathbf{X} \longrightarrow \mathbf{H}$ to a low-dimensional space, e.g. $\mathbb{R}^F \longrightarrow \mathbb{R}^{F'}$, with $F' \ll F$.

First a distance matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is computed, in which each element d_{ij} represents the distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the dataset $\mathcal{X}$. The distance matrix can be view as a weighted adjacency matrix of the graph underlying the node set, and defines an initial edge set $\mathcal{E}$. Manifold learning methods use distance between points to obtain a low-dimensional representation of the data $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$, generally called embeddings, by preserving the manifold $G_M = (\mathcal{V}, \mathcal{E})$'s estimated intrinsic geometry.

This second step can be conveniently framed as an optimization problem, in which the general objective is to find the coordinate vectors of the input data in a lower dimensional space whose dissimilarities are close as possible to the distance d_{ij}. However, manifold learning algorithm can be broadly categorized into two category. The first are the *one-shot* algorithms, which obtain embeddings by using principal eigenvectors of a matrix associated with the distance or neighborhood graph, or by solving some other global optimization problem. The second starts with an initial embedding and then through optimization improve it iteratively [134].

Principal component analysis (PCA) is one of the classical methods for feature extraction, and can be framed as a simple linear form of manifold learning that belong to the first category. PCA applies a transformation of the type $\mathbf{H} = \mathbf{XW}$, in which each vector $\mathbf{x}_i$ in the dataset, is mapped into a new space of uncorrelated variables by using a matrix

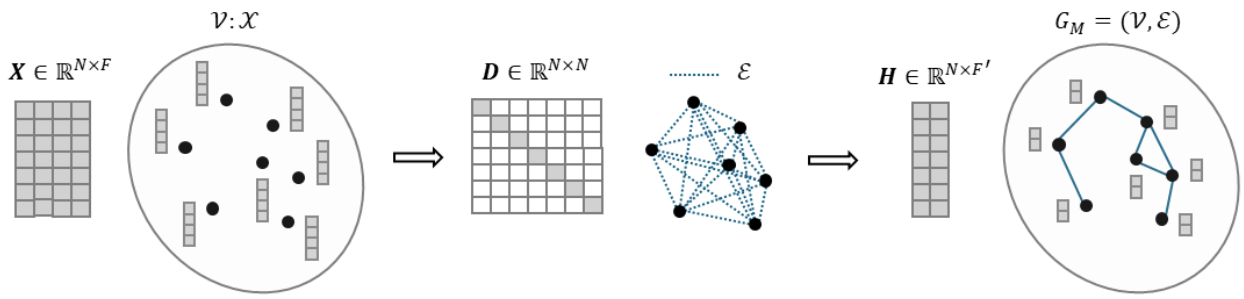


Figure 2.5: Giving in input a dataset $\mathcal{X}$ represented by a feature matrix $\mathbf{X}$, this can be considered in a GRL framework as a point-cloud, which elements belong to a set $\mathcal{V}$ of graph's nodes without edges. Manifold learning methods compute the pairwise distance between the elements of the point-cloud, obtaining a distance matrix $\mathbf{D}$, that can be view as a weighted adjacency matrix identifying an initial set of graph's edges $\mathcal{E}$. Then, learn a low-dimensional representation of the node features $\mathbf{H}$, that best preserves the manifold G_M estimated intrinsic geometry.

$\mathbf{W} \in \mathbb{R}^{F\times F'}$, which columns form an orthogonal basis for the F' orthogonal directions of greatest variance in the data [18]. Specifically, PCA can be associated with the singular value decomposition (SVD) of the dataset matrix, in the form of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^\top$, where $\mathbf{\Sigma}$ and $\mathbf{U}$ represent the matrix of the singular values, and the matrix which columns are the singular left vectors of $\mathbf{X}$, respectively. The solution to the SVD is the transformation that take the form of $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}$, called principal components, consisting in a matrix $\mathbf{H} \in \mathbb{R}^{N\times F'}$, which columns characterize the direction that explain the variance in the data, and are called principal vectors.

One of the first method proposed for non-linear dimensionality reduction is multidimensional scaling (MDS). MDS belong to the first category of manifold learning methods and its classical formulation considers the distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ to be the Euclidean distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2 = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)}, \tag{2.16}$$

while the objective function is called *stress* and can be write as:

$$\underset{\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_N\in\mathbb{R}^p}{\arg\min} \sum_{i\neq j=1}^{N} (d(\mathbf{x}_i, \mathbf{x}_j) - ||\mathbf{h}_i - \mathbf{h}_j||)^2 \tag{2.17}$$

This MDS version is called *metric*, and apply a transformation in a similar way to PCA, in which the matrix $\mathbf{X}$ is derived by the eigenvalue decomposition from $\mathbf{B} = \mathbf{X}\mathbf{X}^\top$. The matrix $\mathbf{B}$ is computed from the distance matrix as $\mathbf{B} = -\frac{1}{2}\mathbf{C}\mathbf{D}^2\mathbf{C}$, with $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{J}_N$ the centering

matrix, $\mathbf{I}$ the identity matrix and $\mathbf{J}_N$ a matrix of ones. MDS can be generally applied to other distance metric in a version called *non-metric*, and through the use of a numerical optimization method is possible to find local minima [116].

However, this is an hard computational problem to solve, making the generalization the main limitation of this algorithm. Isomap is an algorithm that uses the major characteristic of the previous methods, while trying to preserve the intrinsic geometry of the data, by also considering the geodesic manifold distance [210]. Isomap starts by filtering the distance matrix computed with Eq. 2.16, with the creation of a K-nearest neighbors (KNN) graph, in which each point is connected only to the K-top close points, where K is an hyperparameter. Then proceed by computing the distance between the points as the shortest path distance (Eq. 2.14), viewed as a good approximation of the geodesic distance in the manifold, in particular when considering a neighborhood of data sufficiently dense. Finally, Isomap applies to the KNN-filtered distance matrix the classical MDS algorithm to obtain the low-dimensional embeddings.

The second category of algorithms try to overcome some limitations of the previous category, in particular the problem raised is that most of them are not able to include both the local and the global structure of the data when mapping in low-dimension. These algorithms are called also *attraction-repulsion* addressing a balance between the attraction of neighbors in the original space and the repulsion of neighbors in the embedding space [134].

One of most used algorithm that belong to this second category is the t-distributed stochastic neighbor embedding (t-SNE) [127]. This algorithm starts by converting the distance obtained from the high-dimensional space into conditional probabilities that represent similarities. For example, the similarity of the data point j represented by the vector $\mathbf{x}_j$ to the point i with the vector $\mathbf{x}_i$ is:

$$p_{j|i} = \frac{e^{\frac{-d(\mathbf{x}_i,\mathbf{x}_j)^2}{2\sigma^2}}}{\sum_{k\neq i} e^{\frac{-d(\mathbf{x}_i,\mathbf{x}_k)^2}{2\sigma^2}}}, \tag{2.18}$$

expressing the conditional probability that i would pick j as its neighbors, if these are picked under the assumption of a Gaussian probability density, centred in $\mathbf{x}_i$ and with variance σ_i. It is possible to define a similar probability according to Eq 2.18 for $p_{i|j}$, and then compute the joint probability as the symmetrized conditional probabilities $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2}$.

Note that t-SNE is a specific version of the stochastic neighbor embedding (SNE) algorithm [91], in which these conditional probabilities are symmetric. The same consideration be adopted when modeling the conditional probabilities for the points i and j in the low-

dimensional space, and write the symmetric version as:

$$q_{ij} = \frac{(1 + ||\mathbf{h}_i - \mathbf{h}_j||^2)^{-1}}{\sum_{u \neq l} (1 + ||\mathbf{h}_u - \mathbf{h}_j||^2)^{-1}}, \tag{2.19}$$

by considering a Student t-distribution with a single degree of freedom, that has much heavier tails, i.e. it is larger than a Gaussian distribution, and allows to map faithfully from the high-dimensional space. The algorithm proceed by minimizing the Kullback-Leibler (KL) divergence, that is a measure expressing the difference between two probability distributions, between the joint probability distribution in the high-dimensional space P and the joint probability distribution in the low-dimensional space Q:

$$KL(P, Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}. \tag{2.20}$$

One other popular method of manifold learning is Uniform Manifold Approximation and Projection (UMAP) [133]. This algorithm has foundation on topology, a branch of mathematics that studies the properties of geometrical objects that are preserved after continuous transformation.

While the underlying idea behind t-SNE is to match the probability distributions that models the high- and low-dimensional space, with the objective of preserving the probabilities of neighbor similarity, UMAP matches the *fuzzy* (not precise) simplicial sets, defined as a sequence of points with an internal order structure, by preserving the topological structure of the data. UMAP computes the one-directed probabilities $p_{j|i}$ between a point i and its neighbor j as:

$$p_{j|i} = e^{\frac{-d(\mathbf{x}_i, \mathbf{x}_j) - \rho_i}{\sigma_i}}, \tag{2.21}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the notion distance between the points, ρ_i and σ_i are the the local connectivity hyperparameters, set to the distance from i to its nearest neighbor and the local distance around i upon its k-nearest neighbor, respectively.

Then UMAP computes the global probability as $p_{ij} = (p_{j|i} + p_{i|j}) - p_{j|i}p_{i|j}$. The pairwise similarity between points in the lower-dimensional space is defined as:

$$q_{ij} = (1 + a||\mathbf{h}_i - \mathbf{h}_j||^{2b})^{-1}, \tag{2.22}$$

where a and b are hyperparameters which values are set based on the desired minimum distance between point in the embedding space. The loss function minimized through gradient

descent is the cross-entropy loss:

$$CrossEntropy - loss = \sum_{i \neq j} p_{ij} log(\frac{p_{ij}}{q_{ij}}) + (1 - p_{ij}) log(\frac{1 - p_{ij}}{1 - q_{ij}}). \tag{2.23}$$

Biomedical application of manifold learning

Manifold learning has shown significant utility in the biomedical research, for example in the analysis of large omics dataset. Tzeng et al. [217] introduced a scalable method based on MDS and called split-and-combine MDS (SC-MDS), designed to efficiently handle large genomic datasets. This approach reduced computational complexity and improved the reliability of gene clustering, as demonstrated on yeast cell cycle data.

When Isomap become popular, numerous studies dealing with omics data have demonstrated that this algorithm has superior perfomance when compared with PCA and MDS, for example in the analysis of high-density gene expression data [54, 16]. However, MDS still remains one of the reference method for conducting dimensionality reduction when dealing with this kind of data, and recent works apply it to the analysis of chromosome conformation capture (Hi-C) [100] and microbiome data [38].

The second category of manifold learning method have become indispensable tools for visualizing and interpreting high-dimensional biomedical datasets. T-SNE has proven valuable utility in the genomics domain, for example in the visualization of single nucleotide polymorphism (SNP) data, allowing researchers to detect population structure and genetic variation in human datasets [162]. Similarly, UMAP is applied to genomic datasets to detect subtle population structures and phenotypic differences, helping in the identification of hidden subgroups from the UK Biobank populations [204], with detailed links between genetics, location, and phenotypic traits [56].

Beyond omics data, this category of algorithm has proven to be a powerful tool for visualizing complex clinical data derived from EHR, particularly in patient stratification and interpretability of predictive models. For example, t-SNE is applied on deep representations that learned from clinical time series data, revealing distinct clusters that corresponded to different patient mortality outcomes [185], and similarly, this algorithm is applied to data from emergency department visits, facilitating the understanding of patient diversity and giving more interpretable triage strategies [97].

Aligned-UMAP has been introduced as an extension of UMAP specifically designed for longitudinal clinical datasets, allowing for the visualization of patient trajectories over time, and capturing disease progression patterns to enable to stratification of patients large-scale studies such as the Parkinson's Progression Markers Initiative (PPMI) and the Alzheimer's

Disease Neuroimaging Initiative (ADNI) [50].

One recent approach called parametric UMAP, combines UMAP with the optimization employed by NN, by directly learning a parametric relationship between data and embedding [187]. This framework can be used to obtain UMAP embedding in a semi-supervised way, and has successfully employed in computational phenotyping, where an initial phenotype given by clinicians is used as label and compared with the new defined subgroups [65].

2.4 Topological Data Analysis

Topological data analysis (TDA) is a field of mathematics that uses topology techniques to extract information from the underlying shape of datasets viewed as a network [32]. The general input to TDA methods is a set of observations that does not carry any associated topological information, and similarly to manifold learning methods (Section 2.3), the main idea is to connect data points to show a global continuous shape underlying the data. The connections between points are established by adopting a specific notion of distance, or dissimilarity, between them, and for that it is convenient to consider a dataset in input to the TDA methods as a discrete *metric space* or samples of it [37]. This is defined as a tuple $(\mathcal{M}, f_{dist})$ where $\mathcal{M}$ is a set of points and $f_{dist} : \mathcal{M} \times \mathcal{M} \longrightarrow \mathbb{R}_+$ a function for which, given any points x,y,$z \in \mathcal{M}$, the following statements are valid:

I. $f_{dist}(x, vy) \geq 0$ and $f_{dist}(x, y) = 0$ if and only if $x = y$;

II. $f_{dist}(x, y) = f_{dist}(y, x)$;

III. $f_{dist}(x, z) \leq f_{dist}(x, y) + f_{dist}(y, z)$.

However, TDA goes beyond the simple connectivity based on the nearest point, and build a higher-dimensional equivalents of a neighbouring graph, called *simplicial complex*, that intuitively represent a mathematical space as a union of K-dimensional objects, called simplices, defined according to the choose of K, for example for $K = 0$ refers to points, $K = 1$ refers to intervals, and $K = 2$ refers to triangles. Simplices define a simple combinatorial way to describe a topological space, and are used by TDA to translate a discrete set of observations in a suitable topological structure called simplicial complex.

Specifically, it is possible to convert an *open cover*, i.e. a family of open sets whose unions make the whole space, to a simplicial complex, for example by generally forming a Čech complex. Since TDA is proposed for the analysis of large dataset, it works with Vietoris-Rips complexes, that are similar to the Čech complex but only determined by the 0- and 1- simplices and are much easier to compute. In particular, the Nerve theorem

ensure that the created simplicial complexes encode the topology of continuous space in a meaningful way [37].

Three properties make topology interesting for data analysis, and make extracting patterns through shape possible [124]:

- coordinate invariance: since the input to TDA methods is a metric space, the topological construction obtained with TDA does not depend on the choice of coordinate systems but only on the distance function that specify the shape;
- deformation invariance: this property makes TDA methods less sensitive to noise, since they extract key topological features from the data that are robust against small variations or deformation of its shape;
- compression: TDA is able to synthesize the shape of the data while preserving its overall structure, by identifying shapes using simplicial complexes, that act as a kind of compression in which some details can be lost but the important features are kept.

Note that *invariant* is defined in mathematic as the property of an object to remain unchanged after specific transformations or operations.

The previously defined properties make the two TDA main frameworks, that are the *Mapper* algorithm and *persistent homology*, relevant for the analysis of high-dimensional biomedical datasets [197]. Persistent homology acts as a features extractor since characterize the dataset with a vector that quantifies the presence of various topological shapes at different spatial resolutions. Differently, Mapper generates a graph that characterize and summarize the topology of the dataset, providing a tool for data visualization and exploration.

2.4.1 TDA Mapper

TDA Mapper [196] is an algorithm that combines dimensionality reduction techniques with cluster analysis to create a network representation from a high-dimensional data set.

Considering as a starting point a point cloud $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, viewed as a subset of a metric space with a specific function f_{dist} used to measure the distance between the F-dimensional vector, and represented by a feature input matrix $\mathbf{X} \in \mathbb{R}^{n \times F}$, the steps performed by Mapper are presented in Figure 2.6 and specified as follows:

1. compute distance matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ between each pair of points in the dataset, by using the function f_{dist};
2. apply a continuous *lens* $f : \mathbf{X} \longrightarrow \mathbf{Z}$ that project each point in $\mathbf{X}$ to the latent space $\mathbf{Z}$, that is generally simpler, for example with lower dimensionality, and/or meaningful

for the domain of the features in input. Note that the projection is generally performed on the distance matrix **D**, when considering an high-dimensional dataset;

3. divide the projection space with an *open cover* $U = \{u_1, u_2, ..., u_r\}$, that is a collection of open sets whose unions includes **Z**, by using a series of overlapping bins. The number of bins to use for each dimension of **Z** and the percentage of overlap between bins, are usually called *resolution* (r) and *gain* (g), respectively;

4. use a clustering method to perform a clustering step in each of subset of the initial point cloud that falls in a portion that corresponds to an intersection of bins. Mapper creates a graph $G = (\mathcal{V}, \mathcal{E})$, where the nodes are the identified clusters, and the edges indicates the intersection between them, i.e. an edge exists between two nodes if they share at least one data point.

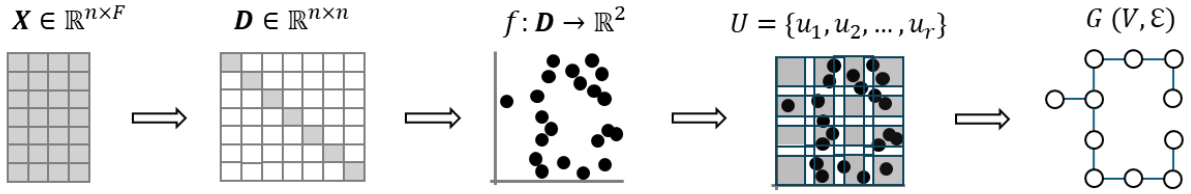


Figure 2.6: The Mapper algorithm receives in input a dataset feature matrix **X**, and first compute the pairwise distance **D** between the dataset's samples. Second, Mapper projects the obtained distance matrix with a lens function f to a low-dimensional representation space, e.g., $\mathbb{R}^2$. Third, it divides the projection with an open cover U by using a series of overlapping bins. Finally, Mapper uses a clustering algorithm to identify the nodes of a new topological graph, where nodes are the clusters and edges express intersection between them.

The output of Mapper depends on the choice of the distance and lens function, the value of the resolution and the gain hyperparameters, and which clustering algorithm is adopted. While the distance function are related to the type of the features in the dataset, the lens functions, the resolution and the gain hyperparameters are generally manually tuned, and the choice of the clustering algorithm is more relaxed.

Biomedical application of TDA mapper

The Mapper algorithm has found wide applications in the biomedical field, in particular for the task of disease sub-typing, i.e. find subgroups of patients within a population, affected by the same disease but with different characteristics. The first application in disease sub-typing involved identifying a previously unrecognized subtype of estrogens receptor-positive breast cancer. By applying Mapper with a lens derived from gene expression deviations from

healthy tissue, Nicolau et al. [149] are abled to uncover a cluster of tumors with a unique molecular signature and favourable prognosis, showing that traditional clustering techniques were not able to lead to the same conclusion.

These results paved the way for the application of Mapper for the analysis of different disease. For example, in studies of type 2 diabetes mellitus (T2DM), Mapper is used to analyse high-dimensional data from EHR, and successfully reveals novel subtypes of T2DM, each associated with distinct clinical profiles and complications [120]. An additional approach is proposed by combining Mapper with pseudo-time series analysis, proposed for investigating temporal phenotypes that defined as sequence of clinical events over time. By using this method it was possible to characterize different trajectories of disease progression, offering insights into how T2DM may develop differently across patients [51].

Mapper has also shown significant promise in cardiovascular (CV), respiratory and brain disease, specifically when dealing with heterogeneous data in input. In aortic stenosis, a Mapper-based approach has identified two distinct progression pathways from mild to severe disease, each characterized by different echocardiographic variables. This not only enhanced diagnostic resolution but also demonstrated how treatment move the patients' positions within the topological network to less severe phenotypic states [33]. Similarly, when studying a population of asthmatic patients, Mapper has helped uncover multiple phenotypic subtypes using diverse data types, from clinical variables to histological and transcriptomic features. These analyses revealed underlying immunological and molecular distinctions among patient subgroups, improving understanding of asthma's complex pathophysiology and suggesting personalized treatment approaches [90]. In the context of traumatic brain injury, Mapper is used to integrate clinical, histological, and genetic data, enabling researchers to identify subgroups of patients with different recovery trajectories and discovering genetic markers linked to poor outcomes in mild cases [150].

2.5 Graph embedding via Neural Networks model

The term embeddings is generally used to indicate a vectorial representation of objects represented by a specific data structure, such as text, image, or simply a tabular dataset. While manifold learning is a method to compute embeddings for the sample in a dataset, the introduction of computational model suitable for graph structures has introduced the possibility to obtain directly embedding for a graph, subgraph, node and edges.

In this section, the focus is on the use of Artificial Neural Networks (ANN) or simply NN, that are a family of learning techniques inspired by the way computation works in the brain, and their adoption for learning embeddings. The basic unit of a NN, resembles the way

information is exchanged in the brain, and is called *neuron*, since receives inputs from the other similar units while computing its own activation. Each neuron performs only simple computations, and to allow for more complex ones they are used in layers, i.e. units that act in parallel.

The way these layers are connected between each other defines a specific architecture of the NN, that always comprise an input layer, an output layer, and one or more *hidden* layers between the input and the output. A NN is said to be *fully-connected* if all the neurons from a layer l have connections with all the neurons from a layer $l+1$, and this is valid for all the NN layers.

Giving an input vector $\mathbf{x} \in \mathbb{R}^n$, a fully connected layer compute a linear combination of the input, that can be described by the following vector-matrix multiplication:

$$h' = \sigma(\mathbf{x}\mathbf{W} + \mathbf{b}), \tag{2.24}$$

where $\mathbf{W} \in \mathbb{R}^{n \times d}$ is the matrix of the learnable parameters, $\mathbf{b} \in \mathbb{R}^n$ a bias vector, σ a non-linear function. One of the most common used architecture is the *feedforward*, also called multilayer perceptron (MLP) [183], where the information flows from the left (input) to the right (output) of the net. With just one hidden layer MLPs has proven to be universal approximators, meaning that they can represent combination of steps function, allowing to approximate continuous function [93].

The term *deep learning* refers to the general employment of a NN architecture with many hidden layers [80]. Considering a NN model made by l layers, its overall computation is defined by $f(x) = f^{(l)}(f^{(l-1)}...(f^1(x)))$. The general objective when developing a DL model is to learn to approximate a function f^*. For example, f^* could map the model input to a categorical label or to a scalar value, if the model is used for classification or regression, respectively. During the *training* of a NN, the computation of the function f, parametrized by the set of parameters θ, is driven to match the desired f^* as close as possible. This is typically achieved by employing stochastic gradient descend (SGD) learning.

2.5.1 Shallow embedding

A first category of graph embedding method proposed in literature is defined with the name of *shallow* embedding. Differently from DL methods, shallow methods involve the transformation of the object of interest with a single layer or in general with simple transformations. In particular, these methods are proposed to learn node or edge embeddings, by optimizing the creation of an embedding space that reflect the closeness of nodes and edges in a graph. Considering for simplicity the task of learning node embeddings, the steps performed by a

general shallow methods are reported in Figure 2.7.

Giving a pair of nodes v_i and v_j, these are first mapped through a learnable function $f : \mathcal{V} \longrightarrow \mathbf{H}$ to embeddings $\mathbf{h}_{v_i}$ and $\mathbf{h}_{v_j}$, respectively, that belong to the embedding space $\mathbf{H} \in \mathbb{R}^{F'}$. Then, the similarity between the nodes $f_v(v_i, v_j)$ and between the embedding $f_{\mathbf{z}}(\mathbf{h}_{v_i}, \mathbf{h}_{v_j})$ is computed, by considering specific similarity functions, for example, f_v can be the distance $d(v_i, v_j)$ of the nodes in the graph and $f_{\mathbf{z}}$ be the euclidean distance. Finally, a loss function $\mathcal{L}(f_v(v_i, v_j), f_{\mathbf{z}}(\mathbf{h}_{v_i}, \mathbf{h}_{v_j}))$ is used to quantify the discrepancy between the computed embeddings and the input similarity in the graph, and it is optimized to reach a minimum value.

By using an *encoder-decoder* framework, is it possible to indicate f as the encoder, i.e. the function that map the objects for which one want to learn embeddings in the embedding space, and the function $f_{\mathbf{z}}$ as the decoder, i.e. the function that quantify structural information about the graph from the learned embeddings [87]. Note that shallow embedding models rely on a function f that constitutes an embedding look-up table encoder.

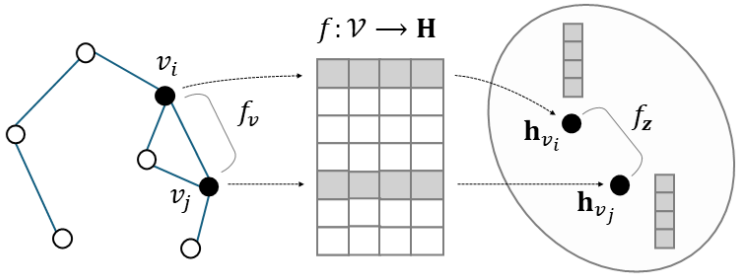


Figure 2.7: Graph shallow embedding models use a shallow encoder f, represented by a simple look-up table, i.e. a learnable matrix of parameters, to embed the graph nodes in $\mathcal{V}$ to the latent space H. This latent space embeds the nodes by taking into account their similarity according to the graph structure, measured by f_v, and compare it to $f_{\mathbf{z}}$ that measures the similarity of the embedding $\mathbf{h}$. The difference between the similarities in the two spaces it is used to compute a loss, which minimization guides the optimization of the parameters during the training.

Different shallow embedding methods can be defined principally according to the choice of the similarity functions f_v and of the decoder function $f_{\mathbf{z}}$. Early approaches work by applying matrix factorization (inner product) as decoder, and indeed manifold learning methods can be included in this category. They consider as similarity between the nodes the adjacency matrix, such as Graph Factorization algorithm [4], or proximity measures like the the Jaccard coefficient adopted by the HOPE algorithm [153], with mean squared error (MSE) is the adopted loss function. In summary, the goal of these methods is to learn node embeddings such that the inner product between the learned embedding vectors approximates some deterministic measure of graph proximity.

In contrast, other methods employ a stochastic measure of graph proximity, measuring the similarity between the nodes as their co-occurrences in sequences of random walks, and use these to create embeddings similarly to Word2Vec [136] with adopting cross-entropy as loss function [161]. Semi-supervised methods, such as Node2vec [82], which combine depth-first and breadth-first search, that are two strategies to sampling nodes in the graph that focus on distant and near nodes, respectively, are employed similarly.

The limitations of the shallow embedding methods reside in the facts that they are inherent trasductive, i.e. they can not generalize when a new node is added in the graph, since the learned look-up table that need to be retrained and additionally, and that they learn embeddings by only considering the graph structure.

Biomedical application of shallow embedding methods

After training shallow embedding models, the learned representations can be used as input for classification models to perform downstream task like classification or regression. With the motivation that biological features are not always available and can be hardly to obtain, the study proposed by Yue et al. [243] evaluates different graph embedding methods on several biomedical link-prediction tasks, such as drug-disease association, and node classification tasks, for example the Unified Medical Language System (UMLS) [21] semantic type classification. The authors reported that graph embedding methods achieved competitive perfomance without relying on biological features, in particular random walk-based method outperform the matrix factorization-based methods.

Nowadays matrix factorization methods are less used, however, they provide an effective method to integrate multiple heterogeneous data sources. For example, Vitali et al. [222] developed a method to compute patient similarities that adopt matrix trifactorization to simultaneously cluster rows (patients) and columns (features or genes) of a dataset, making it especially powerful for subgroup discovery. This approach involves decomposing a large relational matrix, which encapsulates associations between various biomedical entities such as patients, genes, mutations, diseases, and pathways, into three lower-dimensional matrices that capture the latent structure and shared pattern across the different data types. Specifically, the method is first tested on several synthetic dataset and then applied to identify subgroups in acute myeloid leukaemia (AML). The resulting patient representations are showed to be able to identify clinically meaningful AML subgroups with statistically significant differences in survival time.

Since shallow embedding models are proposed to be applied on homogeneous graph, their application on heterogeneous graph structure is not straightforward. To overcome this problem some methods are proposed, in order to take into account the different node and/or

edge semantic types. For example, metapath2vec is one of the most used method when dealing with biomedical networks that have different node types [58]. In brief, this method extend the idea of Node2vec by using paths that leverage some user defined node types, i.e. tailor the embedding with a path made by specific entities of interest. For example, it is possible to consider drugs entities that modify the specific expression of genes entities, that in turn have evidence to cause conditions encoded as disease entities. Zhu et al. [250] have used metapath2vec with the drug-adverse drug reaction-drug-gene-drug entities path, to learn specific graph embedding and enhance the prediction of drug-gene interaction of a matrix factorization method.

2.5.2 Graph Neural Networks

GNNs are a specific architecture of NN that learn embeddings of a graph (and its elements) taking into account both the graph structures and the features of the nodes and/or of the edges. Recalling how a NN projection layer works (Eq. 2.24), and adopting the message-passing framework [78] one general GNN layer computes the hidden representation of a node v_i as follows:

$$\mathbf{h}'_{v_i} = \gamma(\mathbf{x}_{v_i}, \oplus_{v_j \in N_{v_i}} \phi(\mathbf{x}_{v_i}, \mathbf{x}_{v_j}, \mathbf{e}_{v_j v_i})). \tag{2.25}$$

Specifically, a GNN layer first computes the messages between the node v_i and its node neighbors in N_{v_i}, based on their node feature vectors, the edge feature vector, and a possible transformation given by the differentiable function ϕ. Second, it aggregates the messages from all the nodes $v_j \in N_{v_i}$ by using a permutation invariant function $\oplus$, meaning that the output for that function is invariant to the permutation of the elements received in input. Finally, it updates the node features of the node v_i by applying a transformation through the differential update function γ over the aggregated messages and its features.

According to the geometric deep learning blueprint [27], a general GNN architecture is composed of permutation equivariant layers, i.e. shared projection at a node-wise level, and permutation invariant layers, i.e. operator that gives the same output if the element in inputs are permuted.

There are different kinds of GNN layer, while it is possible to define 3 major formulation according to Bronstein et al. [27] and reported in the Table 2.3. The first, that is the one first proposed in literature and links GNN with the way computations are performed by CNNs, it is based on graph signal processing, that is a field that extends traditional signal processing techniques to signals defined on graphs instead of regular domains like time or space. Specifically, this formulation is referred to the *convolutional* formulation, where the convolution operator, that in time domain equals multiplication in the frequency domain,

is defined on graph by considering the multiplication of a graph signal x (a scalar for each node that represent one-dimensional node feature) with a filter g_θ, parametrized by $\theta \in \mathbb{R}^N$ in the Fourier domain:

$$x \star g_\theta = \mathbf{U} g_\theta \mathbf{U}^T x, \tag{2.26}$$

where $\mathbf{U}$ represents the matrix of the eigenvectors of the normalized graph Laplacian, defined as $\mathbf{L} = \mathbf{I}_N - \mathbf{K}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, and defines the graph *frequencies*.

The filter g_θ is a function of the eigenvalues of $\mathbf{L}$, and to approximate the graph filtering, i.e. instead of computing the costly eigenvalues decomposition, Graph Convolutional Networks (GCNs) are introduced. One of the most used GCN model is the one proposed by Kipf et al. [112] and use first-order approximations to obtain a GCN layer as a filter, that propagate and transform node features across neighboring nodes, according to the following formula:

$$\mathbf{H}' = \sigma(\tilde{\mathbf{K}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{K}}^{-\frac{1}{2}}\mathbf{H}\mathbf{W}), \tag{2.27}$$

where $\tilde{\mathbf{A}}$ is the adjacency matrix with self-loops, i.e. each node is connected to itself, and $\tilde{\mathbf{K}}$ the corresponding degree matrix.

Eq. 2.28 can be reported according to the message-passing framework, by considering that the message is computed by considering only the node features of the node neighbors, and these are aggregated with fixed weight $c_{v_j v_i}$, representing the importance of node v_j to the node's v_i representation, while the aggregator operator is chosen to be the summation. The classic GCN layer [112] is reported in Eq. 2.27, where the aggregation over the node neighbor features is weighted by the degrees of the nodes $v_j \in N_{v_i}$ multiplied by the degree of the node v, both measure modified by adding self-loops. After aggregation a transformation is obtained by matrix multiplication with a learnable weight matrix $\mathbf{W} \in \mathbb{R}^{F \times F'}$, where F' is the hidden dimension for the node features.

Formulation	**Layer as message-passing**	**Example of GNN layer**	
Convolutional	$\gamma(\mathbf{x}_{v_i}, \oplus_{v_j \in N_{v_i}} c_{v_j v_i} \phi(\mathbf{x}_{v_j}))$	GCN: $\sigma(\mathbf{W} \sum_{v_j \in N_{v_i} \cup v_i} \frac{\mathbf{h}_{v_i}}{\hat{k}_{v_j}\hat{k}_{v_i}})$	(2.28)
Attentional	$\gamma(\mathbf{x}_{v_i}, \oplus_{v_j \in N_{v_i}} \alpha(\mathbf{x}_{v_i}, \mathbf{x}_{v_j})\phi(\mathbf{x}_{v_j}))$	GAT: $\sigma(\sum_{v_j \in N_{v_i}} \alpha_{v_j v_j} \mathbf{W}\mathbf{h}_{v_j})$	(2.29)
Message passing	$\gamma(\mathbf{x}_{v_i}, \oplus_{v_j \in N_{v_i}} \phi(\mathbf{x}_{v_i}, \mathbf{x}_{v_j}))$	GIN: $f_\theta(\mathbf{h}_{v_i} + \sum_{v_j \in N_{v_i}} (\mathbf{h}_{v_j}))$	(2.30)

Table 2.3: Graph Neural Networks formulation according to the message-passing framework, with equations of the general layer and an example of specific GNN layer.

The second framework is the *attentional*, since use a popular DL mechanism called *at-*

tention. Attention mechanisms were first introduced in sequence modeling tasks such as machine translation, allowing DL models to dynamically focus on the most relevant parts of an input sequence when generating each output [14]. Briefly, attention computes a weighted sum of input features, where the weights, called attention coefficients, indicate the relevance of each input element to a particular query. This idea is formalized by computing attention scores through a function (often involving dot products or NN), applying a softmax to normalize them, and then using these scores to compute a weighted average of the value vectors.

Note that one of the most used attention mechanism is the self-attention [219], that paved the way for the Transformer model. For each word in input in a sequence, three vectors are computed, the *query*, the *key* and the *value* vector, multiplying it with three different weights matrices that are learned during the model training. A score is first computed for each input as the dot product between the query and the key vector, then normalized, and finally scaled by using softmax function. A final multiplication with the value matrix keep the values of the relevant words while minimizing or removing the values of the irrelevant ones. All the procedure can be written in matrix notation as:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\mathbf{k}}})\mathbf{V}, \tag{2.31}$$

and performed multiples times in parallel in order to have multiple representation of the sequence, constituting what is referred to a *multi-head self-attention* layer.

Graph Attention Networks (GATs), translate this idea to graph-structured data, by extending the capabilities of standard GNN to enable nodes to attend over their neighbors with learned, context-dependent weights $\alpha(\mathbf{x}_{v_i}, \mathbf{x}_{v_j})$ through self-attention mechanism, rather than aggregating information with constant weights like in GCN [220]. Specifically, during the computation of the node hidden representation $\mathbf{h}'_{v_i}$, a GAT layer obtain for each edge $e_{v_j v_i}$ connecting v_i with its node neighbors $v_j \in N_{v_i}$, an attention coefficient as:

$$\hat{\alpha}_{v_j v_i} = \sigma(\mathbf{a}^\top[\mathbf{h}_{v_i}\mathbf{W}; \mathbf{h}_{v_j}\mathbf{W}]), \tag{2.32}$$

with σ being the nonlinear activation function $LeakyReLU$, $\mathbf{a} \in \mathbb{R}^{2F'}$ is a learnable weight vector, and $\mathbf{W} \in \mathbb{R}^{F \times F'}$ the learnable weight matrix that multiply the node representation.

To ensure compatibility across different neighbors, the softmax function is applied to normalize the coefficients and to obtain the normalized attention coefficient as:

$$\alpha_{v_j v_i} = \frac{e^{(\hat{\alpha}_{v_j v_i})}}{\sum_{v_q \in N_{v_i}} e^{(\hat{\alpha}_{v_q v_i})}}, \tag{2.33}$$

to finally obtain the GAT layer reported in Eq. 2.29. Here, the aggregation operator is still a summation but now the weights identified by $\alpha_{v_j v_i}$ are feature-dependent. Note that as in [219], multi-head attention can be computed by first executing independent attention mechanism, and then concatenate or average their features.

The last formulation is the message-passing, first introduce by Gilmer et al. [78] as the Message Passing Neural Networks (MPNN), and applied to chemical prediction tasks.

MPNN is the first model proposed to learn features from molecular graphs directly, since it is invariant to graph isomorphism (bijective function that make the exact correspondence between the sets of nodes in two graphs), and it is applied to predict quantum property of molecules from the QM9 dataset [166], reaching highest predictive perfomance when compared with expensive quantum mechanical simulation method. This is formalised in the GNN model by Battaglia et al.[17], as the generalization of the previous published framework, where the notation for the update function (γ), the aggregate operator ($\oplus$) and the message function (ϕ) are introduced.

An example of more complex GNN layer compared to the GCN and GAT is reported in Eq. 2.30, called Graph Isomorphism Network (GIN) layer [237]. GIN is a type of GNN layer designed to match the discriminative power of the Weisfeiler-Lehman graph isomorphism test [230], used to heuristically test the existence of an isomorphism between two graphs. GIN layer applies a sum aggregation, but differently from traditional GNN, applies a learnable projection with a MLP, allowing to capture more effectively the structural information. In addition, GIN can use an hyperparameters that can be learned or fixed, used to weight the influence of the node v_i features.

Note that between the three formulations there is a hierarchy, since the attentional framework can represent the convolutional one by using an attention mechanism implemented as a look-up table, and both the previous frameworks are a special case of the message-passing framework, where the messages are only the sender nodes' features and the sender and receiver's node features for convolutional and attentional GNN, respectively.

However, while the message-passing GNNs are the most expressive, they are not always the most efficient choice. These models involve computing vector-valued messages along edges, which can lead to high memory consumption and increased training complexity. In many naturally occurring graphs, e.g. where the nodes with the same labels are connected, simple aggregation mechanisms tend to perform better, offering advantages in terms of scalability and regularization. GAT provide a useful compromise since capture the interactions within a node's neighborhood while exchanging only scalar attention scores across edges, making them more computationally manageable than full message-passing methods [27].

Biomedical application of graph neural networks models

Differently from shallow embedding models, GNN are generally trained with supervised task such as node or graph classification, making them useful to build predictive model. Considering the task of disease classification, some early works considering to build a graph in which patient is a node represented by specific patient features, such as medical image, with edges indicating the correlation between their phenotypic data, and adopt GNN to make node classification [158]. This data-driven approach has been enhanced since the use of pre-existing knowledge, by creating edges that indicates semantic relation between the dataset variables, and contained in biomedical KB. For example, different works employ GNN to make cancer prediction with gene variables, by using the expression values as feature, and the relations extracted from a PPI network as edges [173, 167].

Recent works combine GNN encoder with KGE decoder, to create a mechanism that can be used when building embedding models, to first pretrain them on biomedical KG, and then continue with the training on specific downstream tasks, by allowing to continue updating the representation learned with biomedical knowledge. For examples, GAT encoder is used in combination with KGE by Alsentzer et al. [9] to build an embedding model called SHEPPERD, suitable for rare disease, based on the PrimeKG KG, and trained on tasks such as casual gene discovery, finding similar patients according to the genes annotation, and disease characterization through patients phenotypes. A very similar pipeline is adopted by et al. [95], to build a graph-foundation model called TxGNN to make zero-shot predictions for therapeutic target.

GNNs are of the most used model when dealing with the task of learning molecule embeddings, since they can leverage the chemicals graph structure where atoms are treated as nodes and bonds as edges [171]. Representing molecules as graphs has improved predictions on quantum-chemistry properties, for example, GNN are successfully used to model interactions among reactants and to predict the outcomes of the reactions [46]. One big employment of GNN is in the prediction of ADME chemical properties, such as solubility, affinity and toxicity, which evaluation need to be of primary interest for the in silico discovery of new drugs [236, 76]. For example, GNN-based models are trained to predict anti-bacterial activity, and are successfully used in the development of antibiotics, by prioritizing molecules with the desired properties [203]. Recently, approach that combine the predictive model capabilities of GNN with the semantic from a KG, and addressing the drug discovery and repurposing have started to emerge [95, 180].

2.5.3 Knowledge Graph Embedding

A series of algorithms for graph embedding is specifically proposed when working with KGs. These method are called Knowledge Graph Embedding (KGE) models, and learn node and relations embedding by taking into account the similarity between the different types of nodes and edges in the heterogeneous graph structure.

Giving a triple (v_i, r, v_j), they represent the entities and the relations in a F-dimensional space, by assigning a value of plausibility to the embedded triple through a decoder function $f_{\mathbf{z}}$. KGE models are trained in assigning a high score to a positive triple, i.e sampled from the KG, and a low score for a negative triple, i.e. a modified version of the sampled triple.

KGE models can be grouped according to different properties [102]. The first categorization can be made by considering the type of representation space employed, e.g. Euclidean [23, 239] or Complex [216, 205] space. Second, the methods differ according to the type of scoring function, that can be generally based on the distance [23] or similarity [239] between the embedding of the elements of the triple. Distance-based scoring function consider the relation r as a translation acting on h and placing it close to t, represented as $h+r \approx t$, while similarity based scoring function compute t with a multiplicative formulation mediated by a projection matrix that is specific for each type of relation, indicated by $h^\top M_r \approx t$, and generally reported as $< h, r, t >$. Third, the methods differ according on which types of relations they can represent, for example 1-to-N relations are the ones that connect an head entity to more tail entities.

Model	**Scoring function**	**Representation space**	**Relation types**
TransE [23]	$-\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^d$	Antisymmetric Inverse, Composite
DistMult [239]	$< \mathbf{h}, \mathbf{r}, \mathbf{t} >$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^d$	Symmetric, 1-to-N
ConvE [55]	$\sigma(vec(\sigma([\mathbf{M}_h; \mathbf{M}_r]\mathbf{W}))\mathbf{t})$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^d$	Antisymmetric Symmetric, 1-to-N
ComplEx [216]	$Re(< \mathbf{h}, \mathbf{r}, \overline{\mathbf{t}} >)$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{C}^d$	Antisymmetric Symmetric, 1-to-N, Inverse
RotatE [205]	$-\|\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|\|$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{C}^d$	Antisymmetric Symmetric,Inverse

Table 2.4: Some of the most used knowledge graph embedding models, divided by the representation space adopted, the scoring function used and which types of represent are able to represent.

In addition, KGE can use shallow encoders or adopt more complex ones, such as 2D CNN

[55], by first reshaping head entity and relation vector to matrix $\mathbf{M}_h$ and $\mathbf{M}_r$, respectively, and then applying convolutional layer before multiplying them to the tail entity embedding. More recent KGE models use GNN encoder[190], by adopting relation-specific learnable layers, to model the directed nature of the KG. Lastly, all these properties make the specific method more or less suitable for specific types or relations that the heterogeneous graph considered holds.

Taking into account all these considerations, some of the most used KGE models are reported in Table 2.4.

Biomedical application of knowledge graph embedding models

KGE models provide an effective ways to capture the biomedical knowledge from a KG, by learning node embedding that can be used for biomedical predictive task. For example, Celebi et al. [34] combine different KB such as DrugBank, KEGG [105], and PharmGKB [232], evaluate different KGE models to learn drug representation, and then use the embedding as input to traditional ML classifiers, such as Logistic Regression (LR) and Random Forest (RF), to make prediction on drug-drug interaction, by showing that KGE provide effective representation for prediction.

However, due to the different ways KGE models are built, there is no direct best choice when choosing one of this model. This is shown, for example, in the findings of Chang et al. [36], where the authors evaluate several KGE models on the SNOMED-CT ontology [198], showing that no single embedding method consistently outperforms others across the considered tasks, highlighting the importance of choosing KGE models based on specific use cases.

KGE can also be used as a quick tool to evaluate the construction of a new proposed KG. For example, Romato et al. [182] computes different KGE models for evaluating the newly proposed Alzheimer KB, then compare them on a classical link prediction task, also called knowledge graph completion. The models that achieved the highest score on link prediction was then used to make inference on the KG created trough a simple drug-repurposing task, aimed at showing the utility of the new knowledge resource, and identifying potential drugs for Alzheimer's disease by cross-referencing them with existing clinical trials.

Recently, KGE models are integrated in a framework called BioBRIDGE [228], that is designed to connect different biomedical foundational models across modalities. By embedding entities from structured knowledge graphs and aligning them with representations from large biomedical language models, BioBRIDGE enhanced relational reasoning and interpretability, demonstrating the power of KGE in harmonizing structured and unstructured biomedical knowledge sources.

Chapter 3

Coronary Artery Disease Computational phenotyping with Topological Data Analysis

3.1 Introduction

Coronary atherosclerosis clinically occurs in acute or stable forms, with syndromes generally included under the term coronary artery disease (CAD). Several scores have been proposed to estimate the sex-specific individual 10-year risk to develop CV disease. However, the accurate identification of at-risk individuals remains a major challenge [77]. Furthermore, adverse clinical events (e.g. heart attack) may occur in a large number of individuals, that are considered as low/intermediate CV risk according by assessing the coronary stenosis [140]. This suggests that several individual factors, not considered during the assignment of the initial phenotype, may contribute to the onset of acute disease-related manifestations. Therefore, there is an urgent need for new stratification strategies, able to integrate different clinical information, to identify more accurate CAD subgroups at an early stage.

TDA is a field of mathematics that uses qualitative geometric features to extract information from the shape of data [124]. Unlike clustering, TDA can recognize related subgroups, providing better insights on how individual samples relate to the whole population [196]. These properties make TDA successful for the analysis of high-dimensional biomedical datasets [149].

Mapper is a TDA algorithm that has reported noteworthy results when applied in electronic phenotyping, in particular for disease sub-groups. For example, Mapper is applied to infer a T2DM patients similarity network in which sub-groups of patients with different

clinical conditions were identified [120], and is also been exploited to identify sub-groups of high- and low-risk aortic stenosis patients using echocardiographic variables as input [33]. However, this method have some limitations that could obstacle its application [197], mainly related to hyperparameters tuning, which can influence both the computational time required and the quality of the output leading to results robustness issues. In addition, none of the available open-source tools is developed with the aim of solving biomedical tasks [218, 209].

In **Aim 1**, a semi-supervised TDA-based framework called *pheTDA* is proposed to perform CAD computational phenotyping on the basis of INTESTRAT-CAD data and domain medical knowledge given by the patients' initial phenotypic definition. pheTDA implements a complete, integrated, pipeline that includes distance matrix calculation, dimensionality reduction techniques for data projection filters selection, and TDA hyperparameters tuning. More in details, pheTDA (i) guides the application of the Topological Mapper algorithm to derive a robust data representation as a topological graph, by considering the initial phenotype; (ii) identifies relevant subgroups of patients from the topology; (iii) assess discriminative features for each subgroup of patients via predictive models.

In the paper presented at the 21st International Conference of Artificial Intelligence (AIME) 2023 [7], this tool is applied on a population of n = 725 patients from INTESTRAT-CAD, by considering only their clinical features. pheTDA identified five novel subgroups, each identified by relevant clinical variables: one of the novel subgroups identified diabetics patients showing a higher CV risk score. In addition, the results obtained are compared with existing clustering algorithms, showing that pheTDA obtains better performance when compared to spectral decomposition followed by k-means.

The entire pipeline is built with open-source Python libraries and publicly available at https://github.com/Sep905/pheTDA.

3.2 Materials and methods

3.2.1 Patients' cohort identification for INTESTRAT-CAD

The study population consists of n = 725 adult subjects enrolled for the Epifania trial. Each patient is associated with a CAD label assigned by the physicians to the CCTA images, based on the amount of atherosclerosis of the coronary that presents the biggest stenosis. This label assignment is performed according to the Coronary Artery Disease-Reporting and Data System (CAD-RADS), that standardize the way the results of a CCTA assigned [49]. The study cohort comprises patients without coronary atherosclerosis (0% degree), patients with sub-obstructive coronary stenosis ($1 \leq degree \leq 69$ %) and patients with obstructive

coronary atherosclerosis ($\geq$ 70% degree). The initial phenotypes Φ is defined as *NOATH*, indicating the patients without coronary stenosis, and *ATH*, for those patients with 1% or more coronary stenosis degree.

Clinical variables are considered for each patient, including medical history (risk factors, family history, comorbidities, symptoms, and drug consumption), examinations (laboratory and instrumental tests) and drug prescriptions. A total of 77 clinical variables, discarding those with at least 10% of missing values are considered. Clinical characteristics of the patients' cohort are reported separately for NOATH and ATH patients, with statistical significance resulting from testing their difference between the initial phenotype in Table 3.1.

Table 3.1: Patients' characteristics divided according to the CAD class. Continue variables are reported as mean (standard deviation) and categorical variables as relative frequency (percentage). Chi-squared/Fisher's exact test or t-test/Mann-Whitney test are used to assess significant difference between the groups for categorical and continuous variables, respectively. P-values are reported in bold if statistically significant (< 0.05).

Variable name	**NOATH (n=287)**	**ATH (n=438)**	**P-value**
Sex, Male	141 (54.4%)	359 (77.4%)	**<0.001**
Family history of cerebrovascular disease	58 (22.7%)	113 (24.8%)	0.594
Family history of cardiovascular disease	123 (48.4%)	248 (54.0%)	0.175
Age, years	54.94 (10.49)	62.16 (9.62)	**<0.001**
Height, cm	170.82 (9.77)	173.35 (9.13)	**0.001**
Weight, kg	73.60 (14.34)	80.81 (15.08)	**<0.001**
Body Mass Index (BMI), kg/m2	25.10 (3.64)	26.82 (4.24)	**<0.001**
Smoking habits			
- Never smoked	159 (61.4%)	230 (49.6%)	**<0.001**
- Former-smoker	74 (28.6%)	137 (29.5%)	
- Current smoker	26 (10.0%)	97 (20.9%)	
Smoking daily quantity, cig/day			
- <10	198 (76.5%)	278 (59.9%)	**<0.001**
- [11-20]	25 (9.7%)	84 (18.1%)	
- >20	36 (13.9%)	102 (22.0%)	
Physical activity frequency			
- Sedentary	32 (12.4%)	113 (24.4%)	**<0.001**
- Occasionally active	134 (51.7%)	215 (46.5%)	
- Active	72 (27.8%)	116 (25.1%)	
- Very active	21 (8.1%)	20 (4.3%)	

Variable name	**NOATH (n=287)**	**ATH (n=438)**	**P-value**
Hypertension	95 (36.7%)	271 (58.5%)	**<0.001**
Diabetes mellitus	9 (3.5%)	53 (11.4%)	**<0.001**
Hypercholesterolemia	120 (46.5%)	272 (58.9%)	**0.002**
Peripheral artery disease	7 (2.9%)	37 (8.5%)	**0.008**
Anaemia	15 (5.8%)	12 (2.6%)	0.050
Chronic Kidney Disease (CKD)	0 (0.0%)	4 (0.9%)	0.329
Thrombophilia	4 (1.6%)	9 (2.0%)	0.931
Angina pectoris	38 (22.8%)	85 (29.5%)	0.146
Arrhythmias	58 (34.9%)	92 (31.8%)	0.565
Atypical chest pain	72 (43.1%)	88 (30.7%)	**0.010**
Non-cardiac chest pain	24 (14.5%)	32 (11.2%)	0.385
Systolic Blood Pressure (BP), mmHg	137.81 (17.57)	143.96 (18.78)	**<0.001**
Diastolic Blood Pressure (BP), mmHg	81.17 (10.55)	82.93 (11.14)	**0.039**
Heart Rate, bpm	72.25 (12.85)	71.06 (12.87)	0.237
Abdominal Circumference, cm	90.61 (12.43)	98.43 (13.85)	**<0.001**
Pelvis Circumference, cm	97.39 (11.46)	102.49 (10.98)	**<0.001**
Oxygen Saturation, %	98.10 (1.40)	97.34 (1.65)	**<0.001**
Years as a Smoker, years	8.80 (12.35)	13.79 (15.26)	**<0.001**
Leukocytes, $10^{-3}/\mu$L	7.48 (1.87)	7.82 (1.84)	**0.017**
Erythrocytes, $10^{-6}/\mu$L	4.86 (0.48)	4.89 (0.46)	0.445
Haemoglobin, g/dL	14.26 (1.42)	14.62 (1.31)	**<0.001**
Haematocrit, %	41.68 (3.68)	42.59 (3.46)	**0.001**
Mean Corpuscular Volume (MCV), fL	86.09 (5.86)	87.24 (6.27)	**0.016**
Mean Corpuscular Hemoglobin (MCH), pg	29.45 (2.36)	30.15 (3.64)	**0.005**
Mean Corpuscular Hemoglobin Concentration (MCHC), g/dL	34.18 (1.08)	34.33 (1.03)	0.066
Red Cell distribution Width (RDW), %	13.01 (1.17)	13.08 (0.99)	0.418
Platelets, $10^{-3}/\mu$L	240.18 (60.51)	233.17 (57.52)	0.124
Mean platelet volume (MPV), fL	10.47 (2.77)	10.89 (8.22)	0.421
Neutrophils, %	60.89 (11.61)	60.89 (9.81)	0.997

Variable name	**NOATH (n=287)**	**ATH (n=438)**	**P-value**
Lymphocytes, %	29.80 (9.57)	29.07 (8.29)	0.283
Monocytes, %	6.99 (2.40)	7.61 (2.24)	**<0.001**
Eosinophils, %	1.92 (4.62)	1.89 (1.45)	0.913
Basophils, $10^{-3}/\mu$L	0.78 (5.70)	0.50 (0.64)	0.281
Neutrophils, $10^{-3}/\mu$L	4.64 (1.82)	4.83 (1.69)	0.157
Lymphocytes, $10^{-3}/\mu$L	2.16 (0.76)	2.22 (0.73)	0.287
Monocytes, $10^{-3}/\mu$L	0.51 (0.19)	0.58 (0.20)	**<0.001**
Eosinophils, $10^{-3}/\mu$L	0.14 (0.32)	0.15 (0.12)	0.625
Basophils, %	0.05 (0.27)	0.04 (0.05)	0.437
Blood sugar, mg/dL	99.82 (19.90)	104.35 (25.76)	**0.015**
Uric acid, mg/dL	4.91 (1.38)	5.37 (1.24)	**<0.001**
Gamma-Glutamil Transferasi (GGT), U/L	26.59 (18.12)	34.96 (27.01)	**<0.001**
Total bilirubin, mg/dL	1.74 (15.44)	0.70 (0.37)	0.154
high sensitivity Troponin I (hsTnI), μg/L	7.80 (58.52)	4.73 (8.92)	0.270
Triglycerides, mg/dL	94.83 (50.21)	108.18 (60.06)	**0.003**
Total cholesterol, mg/dL	195.99 (39.88)	194.65 (34.77)	0.637
High-Density Lipoprotein (HDL) cholesterol, mg/dL	62.83 (17.20)	57.30 (14.87)	**<0.001**
Low-Density Lipoproteins (LDL) cholesterol, mg/dL	120.13 (69.62)	115.60 (30.45)	0.228
C-reactive protein (CRP), mg/dL	1.88 (5.23)	2.17 (4.19)	0.429
High-sensitivity C-reactive protein (hs-CRP), mg/dL	2.05 (5.65)	2.49 (5.72)	0.321
Hemoglobin (Hb)A1c - IFCC, mmol/mol	36.67 (5.09)	39.45 (7.49)	**<0.001**
Hemoglobin (Hb)A1c - DCCT, %	5.49 (0.53)	5.76 (0.69)	**<0.001**
Statins	41 (15.8%)	154 (33.2%)	**<0.001**
New Oral Anticoagulants (NOA)	6 (2.3%)	20 (4.3%)	0.241
Beta blockers	71 (27.4%)	150 (32.3%)	0.197
Angiotensin-converting enzyme(ACE) Inhibitors	43 (16.6%)	93 (20.0%)	0.300
Sartans	34 (13.1%)	126 (27.2%)	**<0.001**
Diuretics	26 (10.0%)	76 (16.4%)	**0.025**

Variable name	**NOATH (n=287)**	**ATH (n=438)**	**P-value**
Nitrates	3 (1.2%)	4 (0.9%)	1.000
Calcium antagonist	16 (6.2%)	70 (15.1%)	**0.001**
Warfarin	0 (0.0%)	5 (1.1%)	0.227
Acetylsalicylic acid (ASA)	38 (14.7%)	142 (30.6%)	**<0.001**
Clopidogrel	0 (0.0%)	12 (2.6%)	**0.021**
Hypoglycemics	7 (2.7%)	47 (10.1%)	**<0.001**
Insulin	2 (0.8%)	11 (2.4%)	0.208
Anxiolytic/Antidepressant	18 (6.9%)	40 (8.6%)	0.516
n-3 PolyUnsaturated Fatty Acids (PUFA)	4 (1.5%)	9 (1.9%)	0.927
Ezetimibe	6 (2.3%)	20 (4.3%)	0.241

The ATH group shows significantly higher proportion of male patients, and higher mean age, when compared to the NOATH patients. As expected, the ATH groups shows higher mean traditional risk factors such as BMI, blood pressure, and pelvis and abdominal circumference, and higher proportion of comorbidities such as hypertension, diabetes mellitus, hypercholesterolaemia. Considering the lab values, the ATH patients reported an higher mean values for haematocrit, MCV, MCH, blood sugar, uric acid, GGT, triglycerides and a lower mean HDL. Lastly, ATH patients have an higher proportion of prescriptions for sartans, diuretics, calcium agonist, ASA, and hypoglycemics.

3.2.2 Experimental setting and pheTDA framework overview

The computational framework implemented in pheTDA and reported in Figure 3.1, is adopted to discover novel subgroups Φ' of CAD patients, giving in input the initial groups $\Phi = \{NOATH, ATH\}$ and the dataset of clinical features.

The framework comprises a step of semi-supervised TDA, where a topological graph is created from the input dataset and an initial phenotypic definition, and a step of computational phenotyping, that introduces a new stratification in the population while characterizing the new subgroups according to the most discriminative features with predictive models. The framework is described in detail in the following sections, followed by the patients' cohort description.

The dataset is splitted into training and test set (70/30 %), using stratified sampling according to the prevalence of Φ. Missing values are imputed with the missForest R package

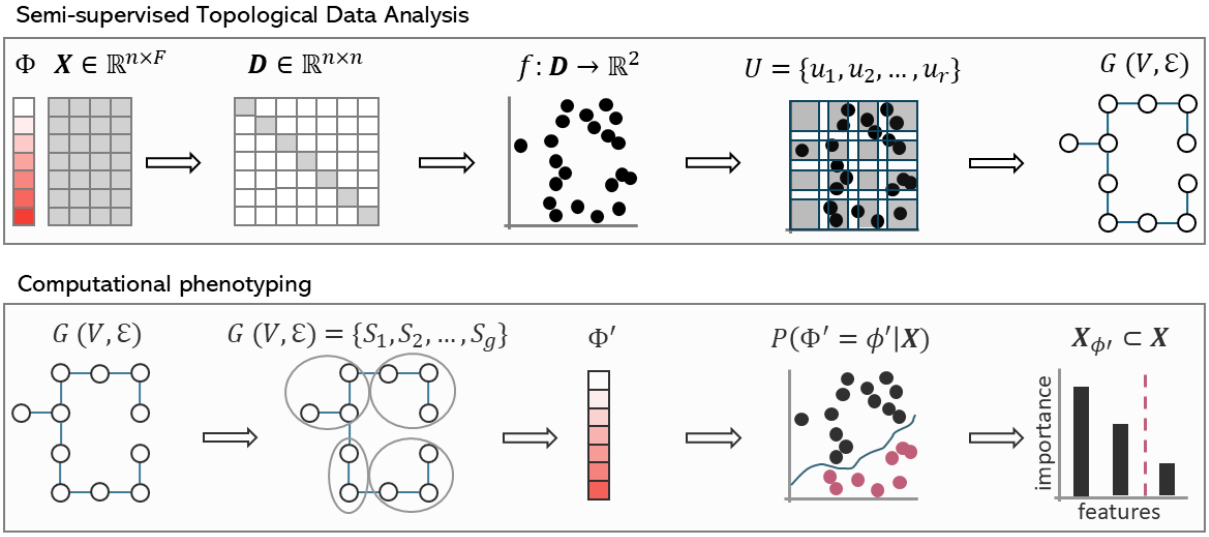


Figure 3.1: PheTDA consists of two main components. The first reported on top, is a semi-supervised TDA pipeline that takes as input a tabular patient dataset and a clinical phenotype Φ. It computes a squared distance matrix, projects it into $\mathbb{R}^2$ space using a lens function, covers the projection with overlapping bins, and applies clustering within each bin intersection. The second component reported in the lower part of the figure, deals with computational phenotyping, which starts from the topological graph produced by Mapper $G(\mathcal{V}, \mathcal{E})$. A community detection it is used to partition the graph and to generate a new stratification Φ'. These subgroups are then characterized by key features extracted from a one-vs-rest predictive model.

[202] and for each patient the sex-specific 10-year cardiovascular risk of occurrence of the first major cardiovascular event (e.g. myocardial infarction) is computed by using the score developed by the CUORE project [156]. Note that the CV risk score is not used as input variable in the pipeline, but for the evaluation of the subgroups identified.

The pheTDA pipeline is employed on the training set to performs hyperparameters tuning, and then applied on the test set to reassign each patient to a novel Φ'. We compared the new partition with standard clustering methods, by evaluating the Calinski-Harabasz index [31], defined as the ratio between the sum of the inter-cluster dispersion and the sum of intra-cluster dispersion.

3.2.3 TDA Mapper semi-supervised hyper-parameters tuning

The input is defined as a tabular patients' dataset $X \in R^{n \times F}$, representing a study population of n patients, each represented by F clinical variables. Each patient is also associated to a label $\Phi = \{\alpha, \beta, ..., \gamma\}$, representing an initial, clinically defined, patient's subgroup, e.g. low-risk (α), medium-risk (β) and high-risk (γ) patients, and defining the initial subgroups

vector $\Phi \in R^n$.

The tool encodes categorical variables with dummy ones and scales numerical variables in the $[0, 1]$ range. According to the features' type, the tool chooses the proper distance metric. In case of only numerical variables the cosine distance is used, while in case of only categorical variables, the Jaccard coefficient is used as distance metric. Since many biomedical datasets are constituted both by categorical and numerical features, pheTDA allows to use the Gower distance [81], to compute the patient distance matrix $D \in R^{n \times n}$.

Gower distance is defined for a pair of example $\mathbf{x}_i$ and $\mathbf{x}_j$ as:

$$d_{ij} = \frac{\sum_{k=1}^{F} s_{ijk} \cdot d_{ijk}}{\sum_{k=1}^{F} s_{ijk}}, \tag{3.1}$$

with s_{ijk} the indicator function that is 1 if feature *k-th* is non-missing for both the samples and 0 otherwise, and d_{ijk} the distance for feature *k-th* between the observations x_i and x_j. This distance is computed as normalized difference if the feature is numerical, as 0 or 1 if the feature is categorical and has the same or different value, respectively, in the samples. Note that in case of ordinal variable, the values are first ranked and then treated as numeric.

After distance computation, the tools executes the TDA Mapper algorithm implemented in the KeplerMapper package [218]. TDA Mapper requires the choose of a *lens* function f_{dist}, the value of the resolution r and gain g hyper-parameters, which define the number of bins and the bins overlap percentage, respectively, and a clustering method C. Both the lens function and the clustering method can have additional hyper-parameters, denoted by f_θ and C_θ, respectively. *pheTDA* defines the set of the Mapper's hyper-parameters as $\theta = f, r, g, C$ and the set of their relative hyperparameters with $\theta' = \{f_\theta, C_\theta\}$.

The tuning of these hyper-parameters is guided by a two-step grid search. The first step involves a grid search over a set of input lens functions, to tune their hyperparameters f_θ. The value of the resolution and gain hyper-parameters and the clustering method are kept constant. The tool, chooses the lens and its hyper-parameters that minimize the graph entropy $H(G)$, defined as:

$$H(G) = \frac{\sum_{i=1}^{N} H(v_i)|v_i|}{\sum_{i=1}^{N} |v_i|}, \tag{3.2}$$

in which node entropy $H(v_i)$ is computed for each node v_i of size $|v_i|$, with respect to the proportion of the initial phenotypic label $P(v_i)$ and reported as follows:

$$H(v_i) = -\sum_{i=0}^{\phi-1} P(v_i) \log_2 P(v_i), \tag{3.3}$$

with ϕ the number of initial phenotypes. Note that the choice of this heuristic allows to obtain a graph in which the nodes are predominantly characterised by individuals belonging to one of the two initials phenotypes.

pheTDA allows the choice of different manifold learning methods as lens, which implementation is taken from publicly available python packages [160]: from simple projection function such as PCA, MDS, and Isomap, to more complex algorithm such as t-SNE, UMAP and parametric UMAP. For parametric UMAP, two functions are implemented through TensorFlow Python package, UMAPnn and UMAPae, which use encoder and autoencoder neural network structures respectively, to parameterize the optimization step performed by UMAP. In addition, pheTDA allows to use *biological lens*, i.e., a user-defined function that is relevant for the task of interest, that is combine with Euclidean distance to form 2D projections.

The second step of grid search is performed on the TDA Mapper hyperparameters r, g, C and its possible hyper-parameters C_θ i.e. to choose resolution, gain and clustering method. Specifically, for clustering method *pheTDA* tried many algorithms from the scikit-learn Python package [160], such as different hierarchical clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [64], and spectral clustering. The optimal combination is based on the evaluation of graph statistics, such as the nodes' size and node's degree distribution since it is relevant to obtain a dense and connected graph in which nodes hardly represent individual patients.

After these steps pheTDA projects the distances in a lower dimensional space $\mathbf{Z} \in R^2$, divides the projection space in a series of overlapping bins in which subsequently performs the clustering step, by using the grid steps results. Each cluster will constitute a node $v \in \mathcal{V}$ in the final graph representation $G = (\mathcal{V}, \mathcal{E})$, while an edge $e \in \mathcal{E}$ expresses that two nodes share at least one sample.

3.2.4 Computational phenotyping with community detection and predictive models

PheTDA identified communities $\{S_1, S_2, ..., S_g\}$ in G through the use of a community's detection algorithm. The tool allows the user to choose from one of the proposed algorithms implemented in NetworkX python package [86]: Girvan Newman algorithm, modularity optimization-based such as Greedy Modularity Optimization and Louvain methods. Note that for Louvain method the default value of its resolution parameters from the NetworkX implementation is used.

While the algorithm identifies a set of communities from the graph nodes, one patient can appear in more than one node in the graph. In this situation, these patients are assigned

to the community that appears most frequently in the nodes which they are associated to, i.e. the majority community. If a patient appears in a set of nodes that does not lead to a majority community, e.g. a patient appears in two nodes, each associated with a different community, the patient is assigned to the community belonging to the nodes with the largest size.

After having introduced in the patients' population the new stratification $\Phi' = \{\alpha', \beta', \gamma'\}$, the framework employs a ML model in a one-vs-rest binary classification setting, in predicting for each patient $n-th$ the probability to belong to each subtype $\phi'-th$, given the patient feature $\mathbf{X}_{n-th}$, defined by $P(\Phi' = \phi'|X)$. In this way, pheTDA learns the predictive models' parameters used to discover the most discriminative features $\mathbf{X}_{\phi'} \subset \mathbf{X}$, for each novel subgroups in Φ'. To fully leverage the visualization strength of the Mapper algorithm, it is possible to use pheTDA to enrich the topology graph obtained, with the most discriminative variables for each community.

pheTDA allows to choose from Elastic Net (EN) [252], RF [25] and eXtreme Gradient Boosting (XGB) [39], that are all models that inherently perform feature selection and allow to inspect feature importance. ML model hyper-parameters are tuned, by maximizing the mean Area Under the Receiver Operating Characteristic curve (AUROC), with a grid search using K-cross validation, in which K is a user-selected parameter. The the ML model with the optimal combination of hyper-parameters is trained on the entire dataset.

3.3 Results

3.3.1 Tune Mapper hyper-parameters with the initial phenotype

The training set composed of patients' clinical variables and their initial phenotype is given in input to the pheTDA pipeline. In particular, five different lens function f are tried on the first step of the grid search, reported in Table 3.2 along with their hyperparameters f_θ and related values. The values in bold identify the best hyper-parameters combination, i.e., the one that leads to the lowest graph entropy $H(G)$. In this phase the Mapper hyper-parameters are kept fixed, in particular resolution and gain values are 18 and 0.5, respectively, and hierarchical agglomerative cluster with complete linkage is used as clustering method.

UMAP is chosen, with *minimum distance* $= 0.9$ and n° *of neighbours* $= 50$ since it leads to the lowest graph entropy of 0.657, and used as input for the second step of the grid search. In addition, pheTDA assists the choice of the lens function by reporting additional plots, some of which are reported in Figure 3.2. For example, the 2D projections obtained for each lens can be inspected and coloured according to the initial CAD phenotype (Figure 3.2A).

Lens function (f)	**Hyper-parameters (θ')**	**Grid search values**	**Graph entropy $H(g)$**
PCA	-	-	0.745
t-SNE	learning rate	[300, **600**, 900]	0.682
	perplexity	[15, 25, **35**, 45]	
UMAP	minimum distance	[0.25, 0.5, 0.75, **0.9**]	**0.657**
	n° of neighbours	[5, 10, 25, **50**, 120, 150, 200]	
UMAP autoencoder	first hidden layer size	[3, **4**]	0.703
	n° of hidden layers	[**200, 400**]	
UMAP encoder	hidden layers size	[**3**, 5]	0.713
	n° of hidden layers	[**100**, 200]	

Table 3.2: Results from the first step of the grid search. Each row reports a lens function f tested, with its hyper-parameters and their values, and the minimum graph entropy obtained for the best combination, highlighted in bold.

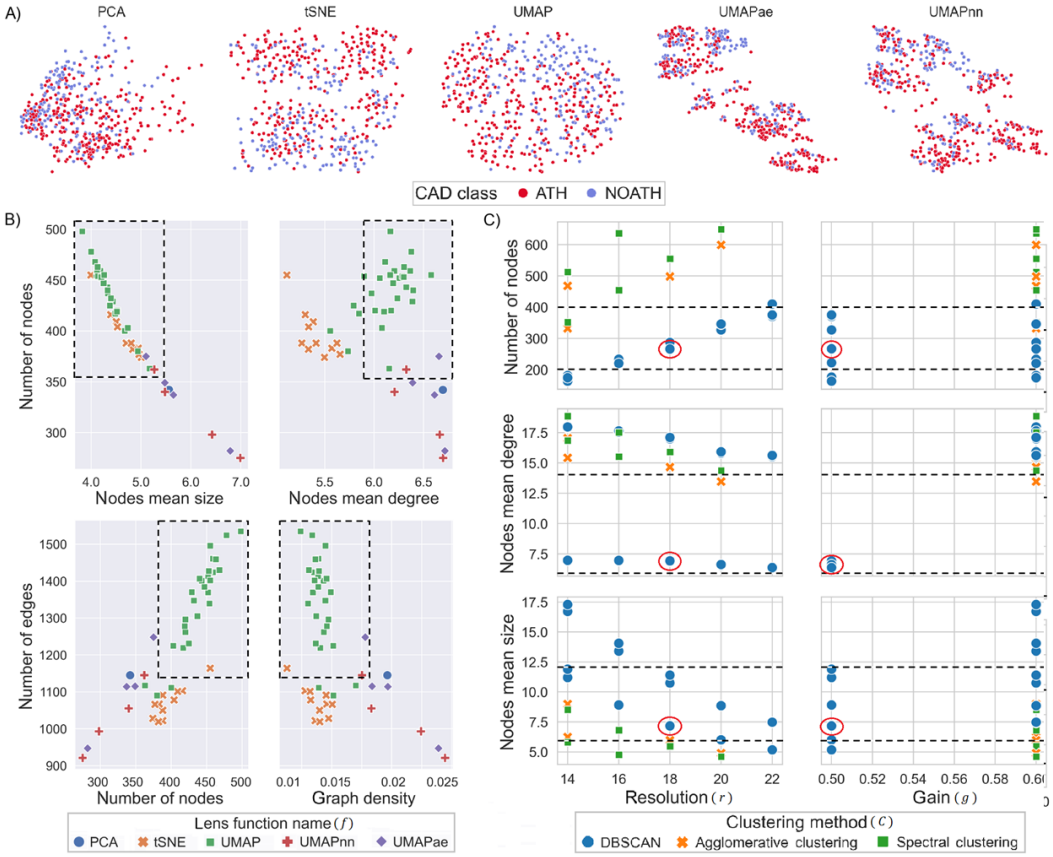


Figure 3.2: Scatterplot of the 2D projections coloured according to the initial CAD phenotype, for the five lenses tested (A); graph statistics plots coloured for the lens and resulting from the first step(B); graph statistics and Mapper hyper-parameters plots, for each clustering method tested (C), and the value chosen is highlighted with a red circle. Note that the grey dashed-line box reported in (B) and (C) indicates the ranges of graph statistics looked when choosing the lens function.

By inspecting graph statistics Figure 3.2B, it is possible to observe in the grey dashed-line boxes, that UMAP projections lead almost always to a topological graph with the highest number of nodes, nodes mean size and degree, number of edges and lower graph density.

After the second step of the grid search, the Mapper hyper-parameters are tuned and the following values are chosen: *resolution* = 18, *gain* = 0.5 and DBSCAN as clustering algorithm, with *epsilon* = 0.5 and *minimum samples* = 2. Again, pheTDA allows to inspect the grid search outputs, by plotting graph statistics vs the values of the Mapper hyper-parameters. Figure 3.2C reports these results, from which it is possible to observe that DBSCAN leads to the lowest number of nodes but to the highest mean size, while *resolution* = 18 keep a balance between the previous metrics and the nodes mean degree. In addition, increasing the gain value makes the average degree increases as well, and decreasing the resolution value increases the number of the nodes while decreases the mean nodes size.

The training set projected in two dimensions with the UMAP lens and the tuned hyper-parameters, divided by CAD group, is presented in Figure 3.3A. This configuration of Mapper hyper-parameters leads to a connected graph, with a mean degree size and node size of 6.9 and 7.2, respectively. The graph is composed by 275 nodes and 1535 edges and is showed in Figure 3.3B.

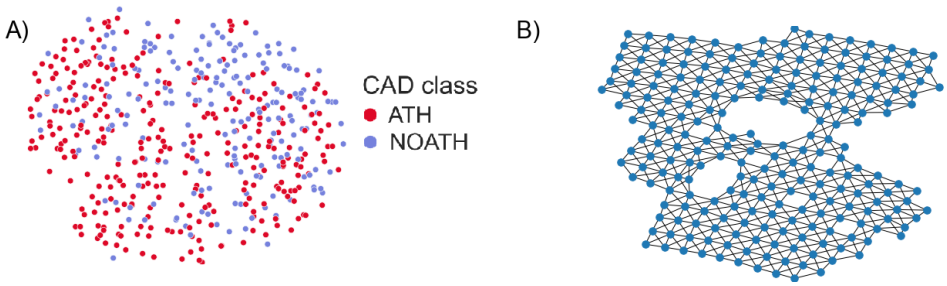


Figure 3.3: Training set 2D projections obtained with UMAP after the first step of the grid search, coloured by CAD class (A); Patients graph generated after the second step of the Mapper parameters grid search (B)

3.3.2 pheTDA identifies communities and characterize them with the most discriminative features

The community detection method used is the Greedy Modularity Optimization algorithm, identifying five subgroups $\Phi' = \{\alpha', \beta', \gamma', \delta', \epsilon'\}$. These are reported in Figure 3.4A, where the topological graph is partitioned according to Φ', coloured by the proportion of the ATH class. In addition, the Sankey diagram presented in Figure Figure 3.4B shows the composition of the patients in the training set according to the CAD class and the new subgroups stratification

Φ'. Subgroups γ' and ϵ' show higher prevalence of ATH patients, while subgroup δ' is mainly composed of NOATH patients. Subgroup α' is only composed of ATH patients.

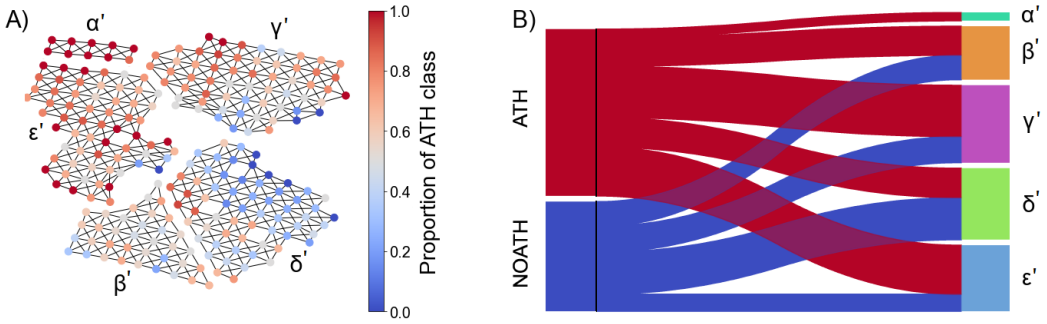


Figure 3.4: Mapper graph partitioned with the communities identified after the application of the community detection algorithm, coloured by the proportion of ATH class (A); Sankey diagram showing the stratification of the new phenotypes Φ' (B).

Model	**Hyperparameters and values**	α'	β'	γ'	δ'	ϵ'
EN	λ_1 = [0.25, 0.5, 0.75] λ_2 = [0.001, 0.01, 0.1, 1, 10]	0.97±0.02	**0.97±0.02**	**0.99±0.01**	**0.98±0.01**	**0.97±0.02**
RF	max. tree depth = [1, 3, 5] min. samples to split = [2, 5, 10] min. samples in a leaf = [1, 5] n° of estimators = [100, 200, 300]	**0.98±0.01**	0.95±0.01	**0.99±0.01**	0.96±0.01	**0.97±0.04**
XGB	gamma = [0, 0.1, 0.2, 0.3] learning rate = [0.1, 0.25, 0.5] max. depth = [1, 3, 5] n° of estimators = [100, 200, 300]	0.97±0.02	0.96±0.02	**0.99±0.01**	0.97±0.01	0.95±0.03

Table 3.3: Results from the computational phenotyping step. Classifier models trained using a one-vs-rest binary classification task with 10-fold cross validation, to predict the patient's membership to each subgroup. For each model, the hyperparameters tuned, the range and the best score (mean and ± AUC), obtained for each subgroup (in bold if the higher for the subgroup), are reported.

After having identified communities of nodes, these are assigned to the patients in the dataset, and prediction models are trained in a one-vs-rest binary classification tasks, while tuning their hyper-parameters for each tasks by adopting 10-fold cross validation and a grid search. The ML models trained, along with the hyper-parameters tried during the grid search, and the validation set mean±std AUC are reported in Table 3.3. EN logistic regression is the model achieving the highest value of AUC, and is retrained on the entire training set. The features that presented an importance coefficient different that zero for each subgroups are reported in the bar plot in Figure 3.5A, along with the hyper-parameters used for the EN trained model.

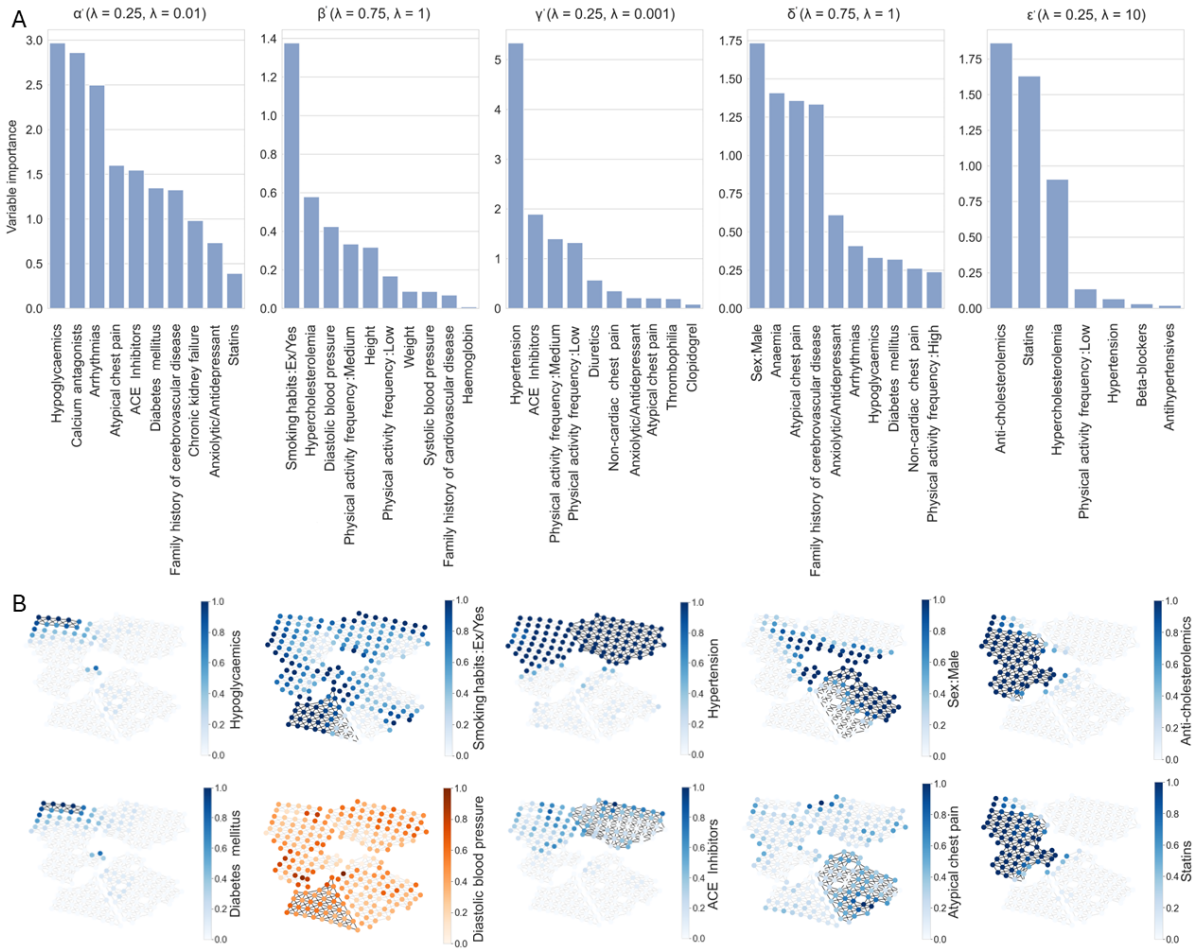


Figure 3.5: Bar plot showing the discriminative features with importance score $\neq 0$ from the trained EN models (A); topological graph enrichment by colouring the communities with the proportion or the mean of their most discriminative features, for categorical or numerical variables, respectively (B).

The most relevant feature for subgroup α' is hypoglycaemia, since it includes the majority of the patients affected with diabetes. These patients show also the highest CUORE CV risk score (0.34), compared to the other subgroup. The smoking habits is the most important feature for the subgroup β', reflecting that the subgroup is prevalently composed of smokers or former-smokers. In addition, the second most discriminative features for β' is hypercholesterolaemia and diastolic blood pressure. The γ' subgroup is characterized by the hypertension variable and the prescription of ACE inhibitors, and shows a medium risk score. Male patients are predominantly, along with cardiovascular symptoms such as atypical chest pain and angina pectoris in δ'. Lastly patients in subgroup ϵ' are the ones prevalently treated with anti-cholesterolemics and statins, since showing a relevant importance of the hypercholesterolaemia while doing physical activity with a low frequency.

By fully leveraing the Mapper algorithm characteristics, the most discriminative features can be inspected by enriching the topological graph, i.e. by colouring the graph obtained according to the variables value. In particular, Figure 3.5B reports the TDA enrichment for some of the most important feature for the new subgroups.

Lastly, the inference capability of pheTDA are inspected, by applying the trained prediction model on the test set. In particular, each EN trained models, i.e. there are 5 models, one for each new subgroups, is used to infer the probability for each patient in the test set to belong to the respective subgroups. The subgroups with the highest probability is assigned to the patient. The stratification obtained on the test set is compared with agglomerative clustering and spectral decomposition followed by k-means, i.e. spectral clustering. pheTDA shows a higher Calinski-Harabasz index (6.14) when compared to spectral clustering (0.59), and a lower score when compared to agglomerative clustering with complete linkage (14.1). The same conclusion can be obtained by first applying the UMAP projection lens, with the optimal combination of parameters found in the first step of the grid search, and then by performing the clustering step.

3.4 Discussion

In **Aim 1** it is presented a TDA-based framework to define novel patients' subgroups, given a dataset of clinical features and the initial clinical phenotypic definition of the patients. This former is used as pre-existing medical knowledge for performing the task of computational phenotyping in a semi-supervised way, guiding the choice of the TDA Mapper hyperparameters. With *pheTDA* five five novel subgroups of CAD patients are identified. The smaller subgroup consists only of ATH patients who represent a subpopulation with diabetes and the highest cardiovascular risk. The other subgroups are more heterogeneous with respect to the proportion of CAD class, but through pheTDA framework, the most discriminating variables for each subgroup are discovered. In addition, when pheTDA is applied on an external test set introduces a more robust partition when compared to spectral clustering.

In the future, pheTDA will be used to investigate how the newly introduced stratification impacts cardiac event prediction. Since at the time of the writing, the INTESTRAT-CAD project was still ongoing, it was not possible to record the presence or absence of cardiac event in patients enrolled. In fact, a limitation of this study is that the dataset is splitted into training and test sets, in order to be able to apply what learned from data and pre-existing medical knowledge on an external dataset. Instead, with the follow-ups provided by the study at hand, the possible steps could be to study how the new stratification, assigned to the baseline visit, is characterised over time, i.e. investigate the trajectories of the patients,

e.g. with survival analysis. In addition, TDA suitable method to discover patient trajectories by using data from multiple time points can be investigated [51].

A possible improvement for the pheTDA framework could be the inclusion of more efficient mechanism for hyper-parameters optimization. For example, Optuna python package [5] implements different optimization algorithm, also belonging to the Bayesian optimization methods, and can be easily integrated in the pipeline. The optimization can be performed on different metrics, i.e. multi-objective optimization, through Pareto Optimization [148], by considering to minimize the graph entropy and the fraction of disconnected nodes (to obtain a connected graph), and to maximize a metric that evaluated the final stratification introduced in the population, such as the Silhouette Coefficient [184].

Considering that most of the projection lens starts from a random seed, and possibly can lead to different projections, that in turn could results in a different topological graph, a possible future investigations could focus on the stability of the entire pipeline. For example, Fitzpatrick et al. [69] use ensemble learning, and particularly their consider to aggregate with a co-occurrence matrix of patients-nodes, many Mapper graphs resulting from the application of several lenses, each view as as a signal detected from the dataset. This study is similar to the application proposed of Aligned-UMAP [50], and makes more stable the application of Mapper; however, it does not address the stability of the Mapper algorithm. This can be evaluated by trying different random seed, and quantifying how much the outputs differ from each other, i.e. the consistency or frustration of the clustering results, with specific metric such as element-centric consistency (ECC) based on PageRank [75].

Lastly, the interpretation of the pheTDA outputs could be enhanced by adopting node-level metrics. These measures can be used to assess the centrality of the nodes in the topological graph, in order to characterize the role of the nodes that acts as bridges between communities [226], which in turn contains patients that are across two subgroups, revealing how the phenotypes are related. In addition, centrality measure can be considered when introducing the new stratification, identifying the most important nodes to which a patients is assigned, if he/she appears in more than one node.

Chapter 4

Combining Clinical and Gene Variables via Knowledge Graph Embedding for Coronary Artery Stenosis Prediction

4.1 Introduction

Risk stratification of coronary artery disease (CAD) is usually accomplished using a set of well-defined clinical risk factors [156, 193]. While traditional risk models remains important, established clinical risk factors for CV disease account for only 50–75% of the variation in major adverse cardiovascular events (MACE) [106]. Notably, around 15–20 % of patients who suffer a myocardial infarction present with just one or two of these risk factors and are not flagged as "at risk" by current predictive models [108]. Considering that CV diseases are largely preventable and can initially present as a fatal event, developing improved strategies to predict risk beyond conventional factors is essential for public health.

The identification of omics biomarkers, for example from RNA-seq, gives a notable objective for developing strategies for diagnosis of CAD, since these are related to the injured tissue [146]. Specifically, peripheral blood gene expression profiling can be used to investigate disease-specific states, study inflammatory responses, and identify biomarkers that reflect the disease severity and activity [103]. Although many new biomarkers have been associated with increased CV risk, few have been rigorously shown to improve upon current risk stratification algorithms by more than a modest margin [77], suggesting that different kind of biomarker should be evaluated and combined.

Recently, predictive models based on ML have been proposed to improve the accuracy of CAD risk prediction. Different approaches can be found in literature, ranging from auto-

mated ML [6] to neural networks [201], using longitudinal data [246] or bigger dataset such as the UK Biobank [3]. Ideally, these methods combine multimodal information sourced from clinical, laboratory, and omics-based data streams.

However, previous works approach the prediction task with classical data-driven method, trying to identifying set of biomarker from large cohort. When dealing with a study population with a relatively small size, leveraging the pre-existing knowledge behind the dataset variables can alleviate the sample size problem. GRL methods are particularly promising for this purpose [62], integrating data-driven approaches with prior biomedical knowledge [121, 104]. Additionally, the increasing availability in medical informatics of biomedical KB harmonized in KG data structure is in favor of adopting an hybrid approach for CAD prediction and risk stratification in precision medicine.

In **Aim 2**, the hypothesis is that KGs can help predictive models when different sources of biomedical features are used in input. For this purpose, the experiments in this study focused on the use of PrimeKG [35], a KG proposed for precision medicine that integrates 20 different biomedical knowledge resources, with the objective of CAS severity prediction for a population of patients from the Epifania trial. In particular, CAS is divided by formulating three different tasks: prediction of mild coronary stenosis (CAS $\geq$ 25 %), prediction of moderated coronary stenosis (CAS $\geq$ 50 %) and prediction of severe coronary stenosis (CAS $\geq$ 70 %).

First, GRL methods suitable for KG, i.e., the Knowledge Graph Embedding (KGE) model, are employed to learn PrimeKG embeddings, and the model with the highest perfomance on the KG completion task is selected. Then, the patients' clinical and gene expression variables are mapped to PrimeKG nodes, in order to contextualize the patients' data with structure medical knowledge, by representing each patient by a combination of KG entities embeddings. Finally, ML classifiers are used to assess the potential predictive power of this knowledge-enriched representation, compared in single-modality settings by using clinical and RNA-seq variables alone, and in multi-modality settings by concatenating the variables. Results shows that: i) the strategy to combining multimodal data generally leads to higher perfomance when compared to the single-modality setting; ii) using KGE embedding to combine the patients variables, achieves better discriminative and calibration performance when compared with classical data-driven techniques.

The code is publicly available and can be found in the following repository: https://github.com/Sep905/ClinGen_viaKG.

4.2 Materials and Methods

Figure 4.1 reports the workflow for the study. First clinical variables and RNA-seq are preprocessed separately and used to train and evaluate the ML model on 10-fold nested cross-validation. These single-source experiments are highlighted by a red and a green arrow, for clinical variables and RNA-seq, respectively. In addition, ML models are trained with the concatenation of the preprocessed variables, as indicated by the dashed blue arrow. Lastly, starting from PrimeKG, first KG representation is learned through KGE models, then each patient's variables is mapped to PrimeKG nodes to represent the he/she as a combination of KGE, and lastly ML model are trained again in multi-modality settings (reported by the solid blue arrow).

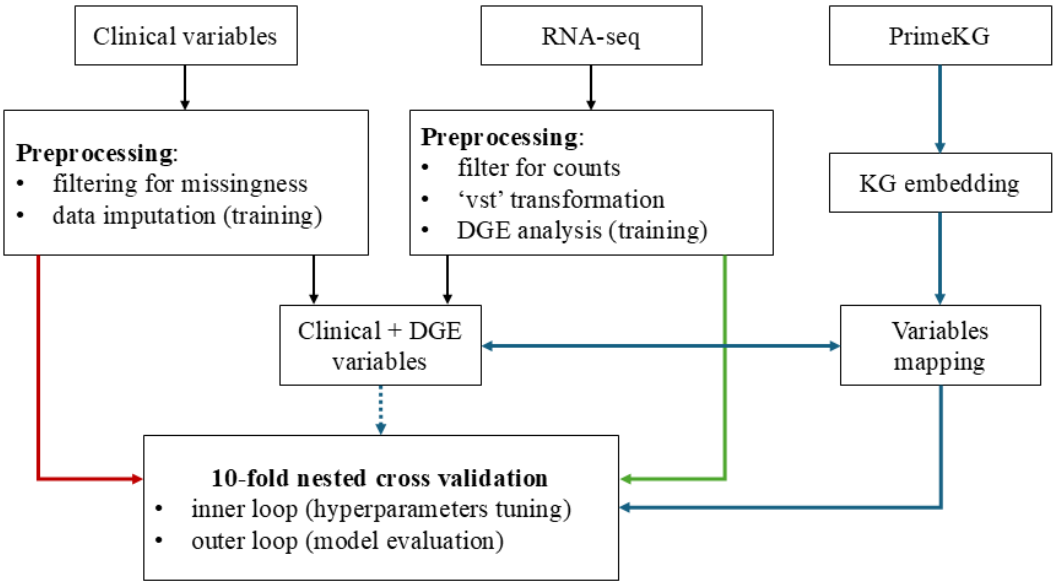


Figure 4.1: Workflow of the study, with arrow coloured by type of experiments: red and green represent the single-modality experiments with clinical variables and RNA-seq, respectively, blue-dashed indicates the concatenation of clinical variables and RNA-seq, while blue-solid the combination of the variables with Knowledge Graph Embedding

4.2.1 Patients' cohort identification and data preprocessing

The study cohort consists of 723 adult subjects from the Epifania trail. Since the study is performed after the one in Aim 1, the cohort is slightly different, considering that some patients patients withdrew consent to use their data in the study. However, the initial CAD phenotype is identified by the percentage of CAS, which distribution is reported in Figure 4.2, for male and female patients separately.

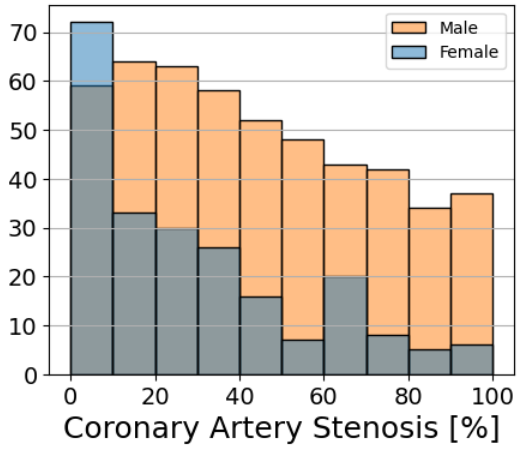


Figure 4.2: Coronary Artery Stenosis (CAS) percentage for the patients' cohort considered, divided according to the sex. While female patients are higher for the lowest value of CAS, as documented in literature CAD is more prevalent in male subjects, that show an higher prevalence for CAS $\geq 10\%$.

Additional clinical variables are considered in the dataset:

- a more detailed medical history, for which previous conditions reported by the patients are mapped and grouped by using the first 3 digits of the International Classification of Diseases, Ninth Revision (ICD-9) codes;
- questionnaires such as the PREDIMED [129], used to assess the adherence to the Mediterranean diet, the Functional Social Support Questionnaire (FSSQ) [26] that measures the perception of a person to the functional support he/she has, and the Patient Healt Questionnaire (PHQ-9) [114] used to measure the depression severity;
- socio-economic variables, such as the level of education, marital status and the type of work.

To constitute the study datasets the clinical variables that show a percentage of missing values $\leq 5\%$ are considered (missForest is used to input the remaining missing values from the training set), resulting in 102 clinical variables, and the matrix of the gene expression obtained after aligning the raw counts against the GRCh38 (hg38) human genome, resulting in a gene expression matrix of raw counts containing 60583 transcripts.

The following data preprocessing is performed (as reported in Section 4.2.4, these steps are performed only on the training set): categorical variables are encoded as one-hot variables, numerical variables are standardized by removing the mean and dividing by the standard deviation, and gene expression variables are preprocessed with DaMiRseq [43]. Specifically, the preprocessing steps include gene filtering, by keeping the ones with a minimum raw

count of 10 for at least 80% of the dataset, applying Variance Stabilizing Transformation (VST), finding and correcting for batch effects by including sex and technical variables, and identifying Differential Expressed Genes (DEGs) that pass the average fold-change (FC) cut off $|FC| \geq 0.5$ and adjusted p-value ≤ 0.05

4.2.2 Learn PrimeKG embedding with Knowldge Graph Embedding models

KGE model	Hyper-parameters	Grid search values
TransE	Loss	Margin Ranking Loss
	loss margin	$[0, 3]$
	regularization	$[\lambda_1, \lambda_2]$
	learning rate	$[0.01, 0.0001]$
ConvE	Loss	Binary cross entropy loss
	learning rate	$[0.01, 0.0001]$
	feature map dropout	$[0, 0.1, 0.2, 0.3, 0.4, 0.5]$
	input dropout rate	$[0, 0.1, 0.2, 0.3, 0.4, 0.5]$
	output channels	$[16, 32, 64]$
	output dropouts	$[0, 0.1, 0.2, 0.3, 0.4, 0.5]$
DistMult	Loss	Margin Ranking Loss
	loss margin	$[0, 3]$
	regularizer weight	$[0.1, 1]$
	learning rate	$[0.01, 0.0001]$
ComplEX	Loss	Softplus
	learning rate	$[0.01, 0.0001]$
RotatE	Loss	Margin Ranking Loss
	loss margin	$[0, 3]$
	learning rate	$[0.01, 0.0001]$

Table 4.1: Knowledge Graph Embedding models implemented in PyKEEN and used to learn PrimeKG embeddings. For each model the hyper-parameters and their values tuned on the validation set are reported.

In this Aim, PrimeKG, a precision medicine-oriented KG, is used as the resource of structure medical knowledge. This KG contains 10 distinct types of nodes, that represent biomedical entities, such as 'disease', 'gene/protein', and 'drug', and 33 edge types, indicating existing knowledge between the entities, belonging to various scales of biological interactions, such as, 'disease' 'is associated with' 'gene/protein' and 'drug' 'targets' 'gene/protein'.

PrimeKG embedding are learned by training and comparing different KGE models [23, 205, 216, 239, 55], by using the PyKEEN Python library [8]. First, PrimeKG triplets (n = 8,100,498) are splitted in 80/10/10 % to create a training, validation and test set of positive triples. Second, according to the PyKEEN default strategy, the model hyper-parameters are optimized with a 10-run random search (KGE models hyper-parameters are reported in

Table 4.1). Each configuration is trained on the training set by keeping the following hyper-parameter values fixed: *embedding dimension* = 200, *training batch size* = 1024, *evaluation batch size* = 64, *number of negative of training triple for positive triple* = 1, *number of max epoch*s = 100, and Adam as optimizer [111].

Note that the evaluation batch size is lower than the training batch size due to the computational limitation of the evaluation protocol. During the evaluation, each triple in a minibatch is corrupted by removing the head, which is replaced with each of the entities in the dictionary in turn (the entities used during the training are excluded), and then corrupted by removing the tail by repeating the process. For each corrupted version a score of dissimilarity is computed to create a list of ranked scores in ascending order. From these, the mean of the reciprocal rank (MRR), which indicates the inverse of the rank for the first correct answer, and the Hitsk, which expresses the percentage of times that a correct answer is at the top-k position in the raked list, are adopted.

4.2.3 Variables mapping to PrimeKG nodes

The dataset variables are manually mapped to PrimeKG nodes, without using imputation strategy, first according to the node types to directly exploit the intrinsic meaning of the variables: past disease or comorbidities are mapped to 'disease' nodes; pre-scribed drugs are mapped to 'drug' nodes when the molecule is known (Statins), otherwise to all the drugs that belong to the same class according to DrugBank; gene expression variables name is first checked by querying NCBI Entrez portal trough Python API then mapped to the 'gene/protein' nodes.

Alternatively, the variables that do not represent directly a node types required some further preprocessing:

- laboratory and other measurements are first identified in the Logical Observation Identifiers Names and Codes (LOINC) terminology [132], and then trough loinc2hpo [245] mapped in phenotype entities from the Human Phenotype Ontology (HPO) [74]. A specific phenotype is assigned if the measurement value is lower or higher (depending on the specific exams) by the normal value/range extracted from [154];

- for the PHQ-9 questionnaire each question is mapped to a specific phenotype in HPO if the answer is positive. However for the FSSQ questionnaire there is no a direct relationships between questions and phenotypes, and the 'Lack of Peer Relationships' is assigned if the total questionnaire score is greater than the mean value. This strategy is used also for the PREDIMED questionnaire, where a series of nutrients, extracted

from [192] and identifying 'exposure' nodes in PrimeKG, are assigned to the patients that have a score greater than the mean.

- the smoking habits are mapped to 'exposure' node such as 'tobacco tar' and 'nicotine', if the patient is a current smoker or a former smoker since at least 15 years (time according to which there are evidence of lowering the cardiovascular risk after smoking cessation [72]).

The variables that are not mapped are the following: Sex, Age, Smoking quantity, Physical activity frequency, Level of education, Marital status and Work. Note that these identify the category of the Social Determinant Of Health (SDOH) variables, and are directly included in the model with their value or categories.

4.2.4 Coronary Artery Stenosis classification settings

Three separate classification tasks are defined by discretizing CAS according to different thresholds, reported a follows with the percentage of positive samples:

- **CAS** $\geq$ **25**%, in which similar to clinical reasoning, the control, defined as the patients with a CAS $\in [0, 24]\%$, are compared with the patients with clinically relevant stenosis, i.e., the patients with a CAS $\in [25, 100]\%$ (61 %);
- **CAS** $\geq$ **50**%, in which the patients with at least a non-obstructive stenosis, defined as CAS $\in [0, 49]\%$, are compared with patients with sub-obstructive or obstructive stenosis, i.e., CAS $\in [50, 100]\%$ (35);
- **CAS** $\geq$ **70**%, where the patients without an obstructive stenosis, defined as the ones with CAS $\in [0, 69]\%$, are compared with the patients that show an obstructive stenosis, i.e., CAS $\in [50, 100]\%$ (18 %).

Each task is performed first by using a single-modality strategy, i.e., by using clinical variables and then DEGs alone, and then with two multi-modality strategies. In the first, the clinical and the DEGs variables are concatenated to fed a ML models, while in the second the patient representation obtained with KG embedding is concatenated to the variables not mapped (Section 4.2.3) and given as input to the ML models.

For each classification task, ML classifiers that automatically include feature selection through regularization mechanism, are trained via the scikit-learn Python package, such as EN and XGB by using a stratified 10-fold nested cross-validation. This strategy includes an outer cross-validation loop with a first training-test split and an inner loop with an additional

training-validation split. For each model, first hyper-parameters tuning is performed on the inner loop, by using a grid search over the model hyperparameters' value (reported in Table 4.2) and selecting the values that maximize the MCC, taking into account all the possible confusion matrix output. The best model configuration is selected and used in the outer loop to evaluate the model discrimination performance on the independent test set, by reporting the AUROC, the Area Under the Precision Recall Curve (AUPRC), the MCC, and the model calibration performance by reporting the Brier score.

ML classifier	**Hyper-parameters**	**Grid search values**
EN	Inverse of regularization strength	$[0.0001, 0.001, 0.01, 0.1, 1, 10, 20, 50, 100]$
	λ_1 ratio	$[0.25, 0.5, 0.75]$
XGB	learning rate	$[1e-2, 1e-1, 0.3]$
	n° of estimators	100
	gamma	$[0.2, 2, 5]$
	n° of columns sample by tree	$[0.25, 0.5, 0.75]$
	tree max. depth	$[3, 5, 7]$
	min. child weight	$[1, 3, 5]$
	λ_1	$[10, 1, 0.1, 0.01]$

Table 4.2: EN and XGB classifiers used to make Coronary Artery Stenosis prediction, their hyper-parameters and values tuned with a grid search.

Note that the preprocessing steps reported in Section 4.2.1 are performed only on the training set for each inner and outer loop and propagated to the variables in the validation or test set.

4.3 Results

4.3.1 PrimeKG learned embeddings and dataset variables mapping

Table 4.3 shows the evaluation results obtained on the test set for the KGE models trained to learn PrimeKG embeddings, where the best (higher) value is in bold for each metric. As expected, TransE and DistMult are the models that performed worse, considering that the first is not able to model 1-to-N relations, i.e., one entity is connected through the same relations to different entities, while the second is not able to model the inverse relations, e.g., a couple of triples in which the relations are different, but the head and entity are permuted. Even if RotatE is not able to mod-el 1-to-N relations, it shows the highest performance when considering all the metrics, and it is chosen as the final model to extract the KGE.

Note that RotatE use an embedding space based on complex numbers, meaning that the method consider both real part and imaginary parts as embedding vectors, doubling the initial embedding size (200). To be able to work with classic ML models, each entity embedding is constituted to be the concatenation of the real and imaginary part, leading to an embedding dimension of 400.

KGE model	Hits@1	Hits@3	Hits@10	MRR
TransE	0	0.03	0.083	0.032
DistMult	0.006	0.029	0.092	0.035
ConvE	0.029	0.047	0.083	0.035
ComplEx	0.122	0.344	0.537	0.267
RotatE	**0.211**	**0.44**	**0.598**	**0.35**

Table 4.3: Knowledge completion results for the trained Knowledge Graph Embedding models on PrimeKG. For each metric the best results is reported in bold.

The PrimeKG entities embeddings are extracted from RotatE model and projected in 2D with UMAP for a qualitatively evaluation. The projections are reported in Figure 4.3, and coloured according to their node type in PrimeKG.

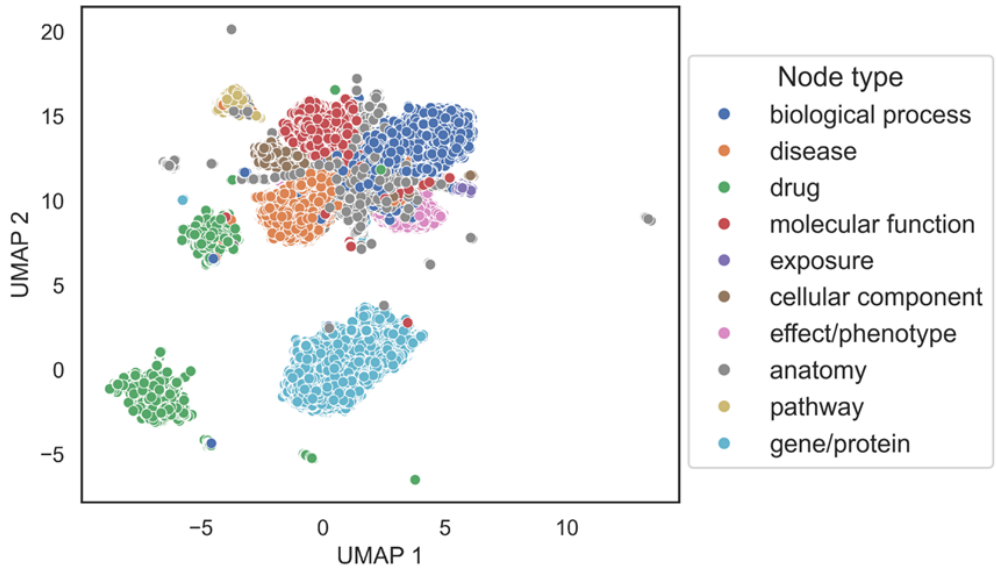


Figure 4.3: UMAP 2D projection for PrimeKG embeddings learned with RotatE, in which each point is coloured by the node type. Note that each initial embedding is the concatenation of the real and the imaginary part of the relative RotatE embedding.

As is notable, the embeddings form clusters by node type: gene/protein, drug and pathway entities are clearly separated in the projection space, while anatomy embedding form a kind of skeleton between the other types, indicating that RotatE can capture the semantics behind the PrimeKG nodes.

In Figure 4.4 is reported a barplot with the number of clinical variables mapped for each patient, grouped by the CAS in a similar wasy the CADRADS system does. From this it is possible to observe that number of mean clinical annotation increases as the CAS increases, reflecting the hypothesis that as CAD severity increases, the patient's state of health is characterised by an accumulation of risk factors and clinical conditions. Note that gene variables are not included, since the same DEGs for each patient in the training set are mapped, with the outputs depending on each fold.

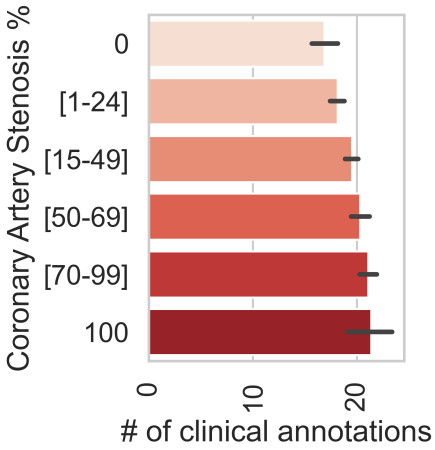


Figure 4.4: Bar plot reporting the number of clinical annotations for patients, grouped according to the Coronary Artery Stenosis. The mean number of clinical annotations increases as the severity of the coronary stenosis increases, indicating that CAS severity is linked with general patient's worsening conditions.

4.3.2 Coronary Artery Stenosis classification

Table 4.4 reports the CAS severity classification results obtained after training (outer-loop) and tuning (inner-loop) the ML models with 10-fold cross validation. The values reported are averaged on the sets of the outer-loop, and reported as mean±std, for each of the three tasks and type of experiments: Clinical(CL) variables, Gene Expression(GE) variables, CL + GE variables and CL + GE variables combine with KGE.

Table 4.4: Coronary Artery Stenosis severity classification results for each binary classification task, obtained by training EN and XGB with 10-fold cross validation (results on the test set), for each experiments: Clinical(CL) variables, Gene Expression(GE) variables, CL + GE variables and CL + GE variables combine with KGE. Each metric is reported with mean±std values. Best values (the highest for AUROC, AUPRC and MCC and the lowest for Brier score) are reported in bold.

Model	**AUROC**	**AUPRC**	**MCC**	**Brier**
CAS $\geq$ 25%				
CL variables				
EN	0.74±.04	0.80±.04	**0.37±.07**	0.21±.01
XGB	0.74±.04	0.80±.04	0.35±.13	0.21±.02
GE variables				
EN	0.66±.04	0.75±.03	0.22±.08	0.23±.01
XGB	0.63±.05	0.72±.05	0.20±.08	0.24±.01
CL + GE variables				
EN	0.73±.03	0.79±.04	0.33±.07	0.21±.01
XGB	0.73±.05	0.80±.04	0.34±.08	0.22±.02
CL + GE variables combined with KGE				
EN	**0.75±.05**	**0.82±.04**	0.36±.11	**0.20±.02**
XGB	0.70±.07	0.77±.06	0.30±.11	0.22±.02
CAS $\geq$ 50%				
CL variables				
EN	0.76±.05	0.62±.06	0.39±.14	**0.20±.02**
XGB	0.75±.06	0.62±.07	0.39±.11	**0.20±.02**
GE variables				
EN	0.62±.05	0.50±.07	0.12±.01	0.24±.01
XGB	0.63±.04	0.51±.07	0.18±.01	0.23±.01
CL + GE variables				
EN	0.76±.06	0.63±.09	**0.40±.12**	0.21±.01
XGB	0.73±.05	0.60±.08	0.32±.11	0.21±.02
CL + GE variables combined with KGE				
EN	**0.77±.05**	**0.63±.08**	**0.40±.09**	**0.20±.02**
XGB	0.73±.05	0.60±.07	0.31±.09	0.22±.01

Model	AUROC	AUPRC	MCC	Brier
CAS $\geq$ 70%				
CL variables				
EN	**0.75±.04**	0.42±.07	**0.32±.08**	0.20±.02
XGB	0.74±.06	**0.44±.08**	0.31±.04	0.18±.02
GE variables				
EN	0.66±.09	0.35±.12	0.17±.13	0.22±.02
XGB	0.70±.07	0.37±.09	0.21±.13	0.18±.02
CL + GE variables				
EN	0.73±.05	0.39±.08	0.27±.10	0.20±.01
XGB	0.71±.04	0.38±.09	0.28±.17	**0.16±.02**
CL + GE variables combined with KGE				
EN	0.73±.05	0.38±.06	0.27±.09	0.20±.02
XGB	0.71±.07	0.41±.07	0.26±.10	**0.16±.02**

Except for the prediction of a CAS $\geq$ 70%, the KG embedding strategy generally improves, albeit slightly, the discrimination and calibration of the ML models used. EN is the best-performing model for the CAS $\geq$ 25% and the CAS $\geq$ 50% tasks, respectively. This can be due to the fact that more complex computational model such as XGB outperforms traditional ML model when trained on bigger dataset respect to the one considered in this study.

In addition, for CAS $\geq$ 25% and CAS $\geq$ 50%, the strategy that uses KG embedding, when inputted to EN leads to higher performance when compared to the strategy that simply concatenates the clinical and gene expression variables. Differently, for the CAS $\geq$ 70%, the single-source strategy that uses the clinical variables reaches the highest metric. This results is not surprising considering that the positive examples for this last task are patients with obstructive CAD for which predicting CAD could be easier and clinical variables already contains a sufficient learning signal.

4.4 Discussion

In **Aim 2**, is investigate the use of PrimeKG to combine clinical and gene expression variables for CAS severity prediction. In particular, first three CAS binary classification tasks are formulated, second different KGE models are trained to learng PrimeKG embeddings,

third the dataset variables are linked to the PrimeKG nodes to create a knowledge contextualized patients representation, that is finally given in inputs to ML models trained for the classification tasks with 10-fold cross validation. The results shows that ML predictive models benefit from the computed knowledge-enriched patient representation through KGE models. In particular, classification results improve when considering KGE for the CAS $\geq 25\%$ and the CAS $\geq 50\%$ tasks, when compared with classic single- and multi-modal strategies to use input features.

The limitations of the proposed approach consist mainly in the hypothesis made for the variables mapping, and in the way not-mapped variables are handled. A possible improvement might be the use of inductive KGE models, that make possible to lear embedding for entities not present in the KG by using GNN encoder, such as NodePiece [70], or state-of-the-art model such as foundation KGE models [71]. In addition, the SDOH variables not mapped open new research avenue for the creation or integration of this kind of variables and their link with medical entities, in new or existing KG, respectively.

The classification results are encouraging, considering that the patients are simply represented as an average of the KGE related to their variables. Possible improvements may be achieved by using Graph Neural Networks (GNNs) as predictive models, considering representing patients as subgraphs of the KG used and formulating the task as graph classification. Specifically, the GNNs mechanism can update the KG entities representation, by following the learning classification signal guided by the gradients, overcoming the use of a fixed-values patient representation. Before making the prediction, a differential pooling layer, e.g., based on self-attention, [118] can be used to create a weighted average of the KGE embeddings, while injecting the learning of the attention weight with the classification signal. In addition, GNNs make it possible to use suitable eXplainable artificial intelligence (XAI) methods, that can be both exter-nal to the predictive model, such as GNNExplainer [240], or already present in the model layer, such as attention mechanism [220], to obtain the relevant portion of the patient subgraph that contributes the most to the model prediction.

For future works, similar classification experiments can be performed by training sex-specific classification models, since there is an unbalance between male and female proportion in the dataset, intrinsic to the characteristics of the CAD. Other additional experiments can be performed by reducing this bias through propensity score matching. When considering this latter strategy, could be also of interest to include the SDOH variables that are showed to be linked with CAD [163]. In addition, considering the objective of the INTESTRAT-CAD project, classification models can be trained in predicting the worsening of the patients that report in the follow-up visits an increased CAS.

Chapter 5

Semantic Knowledge Improves Molecular Machine Learning for Chemical Toxicity Prediction

5.1 Introduction

The scientific community has made numerous contributions advancing computational toxicology, including releasing and maintaining publicly available datasets of *in vivo* and *in vitro* [211, 175], creating informatics KBs containing the scientific evidence related to known toxicity outcomes, such as CTD [53] and AOP-DB [143], and supporting AI research by harmonizing and combining the previously mentioned resources via suitable frameworks such as ROBOKOP [144] and ComptoxAI [181], among others.

While traditional ML has dominated QSAR modeling in the past decades, e.g. RF [206], KNN [247], gradient boosting [244], and deep NN [130, 99], recently the focus has shifted to GRL [199]. The reasons are essentially twofold: GRL models are able to operate directly on graph structures that are a natural and interpretable way to represent molecules, and GRL is suitable for leveraging the multimodal information that resides in biomedical KBs and datasets [121]. Most current research including toxicity predictions adopt exclusively data-driven methods, thus neglecting the wealth of external semantic knowledge resources that have the potential to improve prediction performance [180].

GNN are one of the most used GRL paradigms in computational chemistry [171], and are employed to learn atom, bonds and graph molecules embeddings, but differently from earlier node/graph embeddings methods [4, 82, 23], this kind of model can combine graph structural information (network topology) as well as quantitative features annotated onto

the nodes and edges in the graph. Many works have been published to-date using GNNs in QSAR toxicity predictions, with recent advances including the use of 2D representations of molecules [249, 236], combining molecules' structure with molecular fingerprints via feature concatenation [28] or using them as additional variables to predict in order to define a robust pre-training strategy [48], and using more sophisticated architectures that consider 3D molecular geometry [119], among others.

GNNs are generally considered black-box models, meaning that it is not straightforward to interpret why they make specific predictions, unless XAI methods are added [242]. Explainability is a fundamental step when building a pipeline for predicting molecule properties [231], and is generally assessed by using well-known XAI methods [177], such as Shapley additive explanations (SHAP) [125]. Although new XAI methods focused on toxicity predictions have been proposed in literature [224], little effort has been devoted to the explainability of GNN models for toxicity prediction [168]. Furthermore, in most applications [236, 28, 48], GNN explainability is not assessed with graph ad-hoc explainability methods, such as GNNExplainer. These XAI methods leverage gradient computations and therefore can be used to infer the most informative chemical substructures pertaining to the task of interest.

Considering recent advances in publicly available computational toxicology knowledge resources, the objective of **Aim 3** is to augment a QSAR toxicity prediction model for small molecules combining semantic knowledge between chemicals, gene and Tox21 assays with a KG and GNN. ComptoxAI KG is leveraged to build a dataset of 2D molecular structures and pretrain a GNN encoder module to learn molecule embeddings, a GNN semantic module that update the chemical representation with the semantic knowledge of a heterogeneous graph with chemicals, assay and genes nodes, all of which is then combined into a classification pipeline.

In addition, here the role of adding semantics to toxicity prediction is explored through two parallel approaches: i.) by separately learning molecular embeddings, which are then static in the larger classification model; and ii.) by unifying the GNN encoder and GNN semantic module training, thus dynamically updating molecule embeddings during a single training process. The predictive performance of these strategies are compared with baseline ML and GNN models, and adding semantic knowledge into toxicity prediction models has showed to improve both discrimination and calibration. The analysis are concluded by qualitatively evaluating model explanations by applying GNNExplainer, showing that graph XAI methods can highlight molecule substructures that influence toxicity outcomes.

All code used in this work is available on GitHub at https://github.com/RomanoLab/SemMol and in archived form on Zenodo at https://doi.org/10.5281/zenodo.13946619.

5.2 Material and Methods

In this section, the methods of the study are reported. The framework for the experiments performed is reported in Figure 5.1, and each following subsection describe in details the steps reported in the sub-figures.

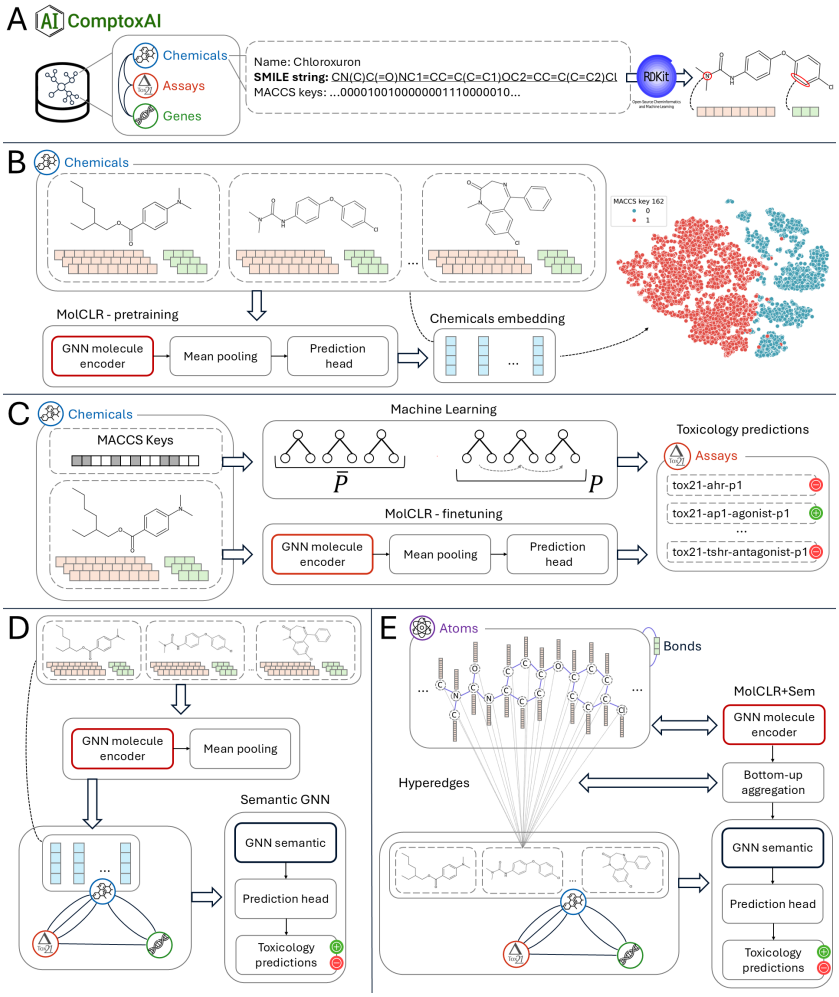


Figure 5.1: Entities and relations of interest are extracted from ComptoxAI (A). SMILES are used with RDKit to generate 2D graph structures for each chemical (B). A GNN molecule encoder is pretrained to learn molecule embeddings reflecting chemical properties (C). ML models with MACCS keys and finetuned MolCLR with 2D graph are used as baselines for toxicity prediction (D). Pretrained embeddings initialize chemical node features in the semantic graph, that is processed by a GNN to update representations and make toxicity predictions (E). Chemical nodes are linked to their atoms; atoms representation is updated, averaged to chemical embedding and passed through a semantic GNN for predictions (F).

5.2.1 Extracting and pre-processing data from ComptoxAI

ComptoxAI is a data infrastructure proposed for computational toxicology, with main component a graph database, resulted from the harmonization of dataset and knowledge resource used in computational toxicology research. The ComptoxAI Python package [179] is used to query the ComptoxAI graph – which contains at the time of writing n = 1,346,793 nodes and n = 10,619,461 edges – to mine information from chemical, assay, and gene entities and the semantic relations between them (Figure 5.1A).

- **Chemicals**: ComptoxAI stores mainly chemical nodes (n = 1,116,847), comprising both approved drug and chemicals substances, belonging to different source such as PubChem [110], DSSTox [83], DrugBank [113], CTD [53]. Most of the chemicals are small molecules and have a molecular weight < 500 Da. From the ComptoxAI chemicals entities are extracted the name, the canonical Simplified Molecular Input Line Entry System (SMILES) string [229], that is one of the most popular way to represent a molecules as a ASCII string made by molecules atoms, and the MACCS 166 fingerprint [61] as chemical features, consisting of a bitstring in which the values 1 and 0 indicate the presence/absence of a specific molecular structural component, respectively, leading to a set of n = 1,078,384 chemical substances. An example of chemical extracted and its attribute from the graph is reported in ported in Figure 5.1A.

- **Assays**: The toxicologic assays available in ComptoxAI derive from the Tox21 datasets [211], constituted by a set of screening results performed in vitro to test different toxicity endpoints of chemicals, on specific cell line. ComptoxAI include $\sim$ 70 Tox21 assay nodes, each of them comprises a set of active and inactive chemicals. In the experiment are included only the assays in which the number of active chemicals is at least the 1% of the sum between active and inactive chemicals (n = 37). The assay included belong to the following assay category: nuclear receptor (NR), stress response (SR), cytotoxicity (CT), gene toxicity (GT), developmental toxicity (DT) and G protein-coupled receptors (GPCR). The number of chemicals with indications for each assay range from n = 5,388 to n = 7,748 and the percentage of positive sample range from 1.58% to 24.84%.

- **Genes**: The gene entities included in the ComptoxAI come from NCBI genes [188] and are n = 193,313. Each gene node could be connected to one or more chemical nodes, depending on whether its expression is regulated by certain chemicals or if some chemical substances bind to it, with evidence extracted from AOP-DB [143] and

Hetionet [89]. In addition, Hetionet is also used in ComptoxAI to derive information about interactions between genes, i.e., edges between gene nodes. Lastly, gene entities are connected to Tox21 assay nodes, indicating the toxicology target.

5.2.2 Pretraining GNN on ComptoxAI chemicals

Starting from the set of the extracted ComptoxAI chemicals, a dataset is created to pretrain a GNN model and learn molecule embeddings. For each chemical, the 2D structure representation is obtained, computed using the RDkit Python package [170] from the chemicals' canonical SMILES representation. Each resulting chemical is represented in the dataset as an undirected graph $G_{c_i} = (\mathcal{A}, \mathcal{B})$, characterized by $\mathcal{A}$ the set of the $N_{\mathcal{A}}$ nodes, representing the atoms, and $\mathcal{B}$ the set of the edges, representing the bonds (Figure 5.1A).

Atom Property	**Description**	**Possible Values**
Atomic number	Number of protons in the atom nucleus.	$[0^*, 118]$
Chirality	Property indicating if the atom cannot be superposed on its mirror image by any combination of rotations, translations and some conformational changes.	[0: 'CHI_UNSPECIFIED'*; 1: 'CHI_TETRAHEDRAL_CW'; 2: 'CHI_TETRAHEDRAL_CCW'; 3: 'CHI_OTHER'; 4: 'CHI_TETRAHEDRAL'; 5: 'CHI_ALLENE'; 6: 'CHI_SQUAREPLANAR'; 7: 'CHI_TRIGONALBIPYRAMIDAL'; 8: 'CHI_OCTAHEDRAL']
Degree	Number of directly-bonded neighbors for an atom. Independent of bond orders but dependent on hydrogen representation.	$[0, 11^*]$
Formal charge	Hypothetical charge assigned assuming equal electron sharing in bonds. Not the real atomic charge.	$[-5, 7^*]$
Number of hydrogens	Total number of implicit and explicit hydrogens. Implicit hydrogens are not stored as atoms; explicit ones are.	$[0, 9^*]$
Number of radical electrons	Number of unpaired electrons.	$[0, 5^*]$
Hybridization type	Mixing of atomic orbitals to form hybrid orbitals for bonding, per valence bond theory.	[0: 'UNSPECIFIED'*; 1: 'S'; 2: 'SP'; 3: 'SP2'; 4: 'SP3'; 5: 'SP3D'; 6: 'SP3D2'; 7: 'OTHER']
Is aromatic	Whether the atom is aromatic.	[0: False, 1: True]
Is in ring	Whether the atom is part of a ring structure.	[0: False, 1: True]

Table 5.1: List of properties (features) extracted trough RDkit and used to represent each atom in the molecules pretraining dataset. Each feature is listed with its name, a short description and the values that can assume. Values with (*) represent the mask values used during the pretraining of the GNN encoder.

Each node is represented as a 9-dimensional array of atom features, reported in Table 5.1, with their name, a description and the range of possible values. By stacking the atom features it is possible to define the nodes features matrix $\mathbf{X}_{\mathcal{A}} \in \mathbb{R}^{N \times 9}$. Note that the values

reported with an asterisk (*) are the ones used as mask value during the pretraining of the GNN encoder.

Edges are represented as a 3-dimensional array of bond features, reported in Table 5.2, with their name, a description and the range of possible values. By stacking the edges features it is possible to define the edges features matrix $\mathbf{E}_{\mathcal{B}} \in \mathbb{R}^{M \times 3}$. Note that self-loop edges, that are represented by the bond type 22, and defined as a connection of a node with itself, are added for each atom during the pretraining of the GNN encoder.

Edge Property	**Description**	**Possible Values**
Bond type	Type of bond between two atoms.	[0: 'UNSPECIFIED'; 1: 'SINGLE'; 2: 'DOUBLE'; 3: 'TRIPLE'; 4: 'QUADRUPLE'; 5: 'QUINTUPLE'; 6: 'HEXTUPLE'; 7: 'ONEANDAHALF'; 8: 'TWOANDAHALF'; 9: 'THREEANDAHALF'; 10: 'FOURANDAHALF'; 11: 'FIVEANDAHALF'; 12: 'AROMATIC'; 13: 'IONIC'; 14: 'HYDROGEN'; 15: 'THREECENTER'; 16: 'DATIVEONE'; 17: 'DATIVE'; 18: 'DATIVEL'; 19: 'DATIVER'; 20: 'OTHER'; 21: 'ZERO'; 22: 'SELF LOOP']
Stereo	Indicates the spatial arrangement of the bonds in 3D.	[0: 'STEREONONE'; 1: 'STEREOANY'; 2: 'STEREOZ'; 3: 'STEREOE'; 4: 'STEREOCIS'; 5: 'STEREOTRANS']
Is conjugated	Whether the bond is conjugated, indicating the overlap of a p-orbital with another across an adjacent bond.	[0: False; 1: True]

Table 5.2: List of properties (features) extracted trough RDKit and used to represent each edge in the molecules pretraining dataset. Each feature is listed with its name, a short description and the values that can assume. Note that the bond type 22, indicating self-loops, is added during the pretraining of the GNN encoder to each atom.

Molecular Contrastive Learning (MolCLR) is used as framework for pretraining the GNN model [227]. Based on SimCLR, the framework uses self-supervised learning to train a GNN encoder with the normalized temperature-scaled contrastive loss (NT-Xent) used to maximize the agreement between augmented versions of a training sample (defined as positive pairs) encoded with a GNN model [40].

Specifically, given a training example from a minibatch of N samples, MolCLR first creates two augmented version G_{c_i} and G_{c_j} by perturbing through node masking or subgraph deletion. Second, a GNN encoder is applied to update node and edge embeddings by obtaining graph embedding $\mathbf{h}_{c_i}$ and $\mathbf{h}_{c_j}$ via pooling node features, and then a projection head modelled as a MLP obtains the latent representation $\mathbf{z}_{c_i}$ and $\mathbf{z}_{c_j}$ (Figure 5.1B). Finally, the

NT-Xent loss is applied to each pair of positive samples (G_{c_i}, G_{c_j}):

$$\mathcal{L}_{G_{c_i},G_{c_j}} = -log\frac{\frac{exp(sim(\mathbf{z}_{c_i},\mathbf{z}_{c_j})}{\tau}}{\sum_{t=1}^{2N}\mathbb{1}\{t \neq i\}\frac{exp(sim(\mathbf{z}_{c_i},\mathbf{z}_{c_j})}{\tau}}, \tag{5.1}$$

where sim is the cosine similarity between the two vectors, i.e. the dot product of the vectors divided by the product of their magnitude, τ is the temperature parameter, and $\mathbb{1}\{t \neq i\}$ is equal to 1 if $t \neq i$, otherwise is 0. Note that the framework does not create negative samples but instead use the augmented versions of the other molecules in the training minibatch.

MolCLR's authors official code is used – implemented using the PyTorch Geometric (pyg) package [68] – to implement the pretraining framework. This implementation uses a GIN [237] encoder (Eq. 2.30) – one of the most widely used GNN encoders for molecules – with a modification that allows edge embeddings [94]. This GNN layer uses a MLP as an update function and sums neighbor features by applying an aggregator operator to edge embeddings, followed by a batch normalization layer. ReLU is used as an activation function for all GNN and linear layers. As a graph augmentation strategies, the atom masking and bond deletion are adopted, meaning that for each molecule 25% of the atoms are hidden by replacing their features with special value token (mask) and 25% of the bonds are deleted from the molecule graph. The model's block diagram is shown in Figure 5.4a.

The pretraining chemical dataset is splitted into training (95%) and validation (5%) sets, the model is trained for 100 epochs using the Adam optimizer and cosine annealing learning rate scheduler [123], and the model is evaluated by calculating validation loss at the end of each epoch. To verify the embeddings capture meaningful information about chemical function, these are projected into two dimensions using t-SNE and coloured according to their chemical and physical properties.

5.2.3 QSAR modeling and baseline models

QSAR modeling is defined as building a predictive model of the form:

$$\hat{y} = f(c) + err, \tag{5.2}$$

where c is a set of chemicals, traditionally described by molecular structure descriptors, f is the computational model to be trained, err is the prediction error made by the model and $\hat{y}$ is the measure of activity to be predicted, e.g. toxicity. The set of Tox21 toxicologic assays extracted and filtered from ComptoxAI is used as dataset for the QSAR toxicity prediction, and is reported in Table 5.3. Each sample in these datasets consists of $\{c_i, y_i\}$, with c_i a

chemical substance and y_i the dataset label, with value $y_i = 1$ or $y_i = 0$ if the chemical is respectively active or inactive for the assay, by treating each toxicologic assay as a binary classification task.

Each dataset is splitted in training/validation/test with a proportion of 70/15/15. For each model, parameters learning (training) is performed on the training set, hyperparameters tuning on the validation set, by choosing the hyperparameters' values that maximize the AUROC, and model evaluation to assess model generalizability on the independent test set, by considering both discrimination performance (with AUROC, AUPRC, and MCC metrics) and calibration (with Brier score). The splitting procedure is repeated five times with a random initialization to compute average performance metrics, comparing them between models by averaging the mean values obtained for each assay and applying paired t-test, and finally correcting for the number of pairwise comparison between the classifier with the Holm-Sidak correction.

Table 5.3: List of Tox21 assay used for toxicity prediction. For each assay the table reports the name, the dataset name, the target category, the cell line and the number of compounds with the percentage of active ones (table continues in the next page).

Assay and Tox21 Name	**Target Category**	**Cell Line**	**Cell Type**	**Total Compounds (% Active)**
Aryl Hydrocarbon Receptor (AhR)	NR	HepG2	Liver	6,991 (11.59%)
Activator Protein 1 (AP-1) Agonist	SR	HepG2	Liver	6,860 (8.34%)
Androgen Receptor (AR) BLA Agonist	NR	HEK293	Kidney	7,245 (3.48%)
Androgen Receptor (AR) BLA Antagonist	NR	HEK293	Kidney	6,556 (7.38%)
Androgen Receptor (AR) MDA Agonist	NR	MDA-kb2	Breast cancer	7,748 (3.55%)
Androgen Receptor (AR) MDA Agonist (with Antagonist)	NR	MDA-kb2	Breast cancer	7,211 (2.65%)
Androgen Receptor (AR) MDA Antagonist	NR	MDA-kb2	Breast cancer	6,505 (5.66%)
Androgen Receptor (AR) MDA Antagonist (lower Agonist)	NR	MDA-kb2	Breast cancer	6,209 (14.38%)
Antioxidant Response Element (ARE)	SR	HepG2	Liver	5,988 (16.53%)
Aromatase	SR	MCF-7	Breast cancer	6,646 (11.01%)
Constitutive Androstane Receptor (CAR) Agonist	NR	HepG2	Liver	6,864 (12.98%)
Caspase-3/7 (CHO)	CT	CHO	Ovary	7,405 (2.38%)
Caspase-3/7 (HepG2)	CT	HepG2	Liver	7,294 (5.29%)
Enhanced Level of Genome Instability Gene 1 (ELG1; human ATAD5)	GT	HEK293T	Kidney	7,560 (3.44%)
Estrogen Receptor (ER) BLA Agonist	NR	HEK293	Kidney	7,475 (5.34%)
Estrogen Receptor (ER) BLA Antagonist	NR	HEK293	Kidney	6,588 (5.27%)
Estrogen Receptor (ER) BG1 Agonist	NR	VM7Luc4E2	Breast cancer	6,482 (10.86%)
Estrogen Receptor (ER) BG1 Agonist (with Antagonist)	NR	VM7Luc4E2	Breast cancer	7,315 (1.61%)

Assay and Tox21 Name	**Target Category**	**Cell Line**	**Cell Type**	**Total Compounds (% Active)**
Estrogen Receptor (ER) BG1 Antagonist	NR	VM7Luc4E2	Breast cancer	6,744 (5.44%)
Estrogen Receptor (ER) BG1 Antagonist (lower Agonist)	NR	VM7Luc4E2	Breast cancer	6,546 (6.99%)
Estrogen Receptor Beta (ERβ) Antagonist	NR	HEK293	Kidney	6,246 (7.41%)
Thyroid Receptor Beta (TRβ) Antagonist	NR	GH3	Rat Pituitary	5,827 (5.69%)
Glucocorticoid Receptor (GR) BLA Agonist	NR	HeLa	Cervical cancer	7,211 (2.45%)
Glucocorticoid Receptor (GR) BLA Antagonist	NR	HeLa	Cervical cancer	6,383 (5.56%)
Histone Deacetylase (HDAC)	GT	HepG2	Liver	7,330 (4.29%)
Heat Shock Response Element (HSE) BLA	SR	HeLa	Cervical cancer	6,643 (6.11%)
Mitochondrial Toxicity	SR	HepG2	Liver	6,476 (18.61%)
Peroxisome Proliferator-Activated Receptor Gamma (PPARγ) Antagonist	NR	HEK293	Kidney	6,117 (5.95%)
Progesterone Receptor (PR) BLA Agonist	NR	HEK293	Kidney	7,380 (1.58%)
Progesterone Receptor (PR) BLA Antagonist	NR	HEK293	Kidney	6,431 (12.86%)
Pregnane X Receptor (PXR) Agonist	NR	HepG2	Liver	6,534 (24.84%)
Retinoic Acid Receptor (RAR) Agonist	NR	HEK293	Kidney	6,259 (7.23%)
Retinoic Acid Receptor-Related Orphan Receptor (ROR) Antagonist	NR	HEK293	Kidney	5,388 (9.13%)
Retinoid X Receptor (RXR) BLA Agonist	NR	HEK293	Kidney	5,924 (3.17%)
Sonic Hedgehog (Shh) Antagonist	DT	NIH3T3	Murine embryo fibroblast	5,701 (16.54%)
Thyroid Stimulating Hormone Receptor (TSHR) Agonist	GPCR	HEK293	Kidney	7,016 (4.32%)
Thyroid Stimulating Hormone Receptor (TSHR) Antagonist	GPCR	HEK293	Kidney	7,155 (3.51%)

Traditional ML models and finetuned MolCLR model are used as computational model for toxicity predictions. Specifically, for ML, two of the most used ML model are used, i.e. RF and XGB, and as molecular structural descriptors the MACCS 166 fingerprint are considered (Figure 5.1C). For each assay the model hyperparameters are tuned by creating different models' configuration with grid search, and by using the hyperparameters' range of values used in Romano et al. [180]. The scikit-learn Python package [160] is used to implement the ML models, by weighting the samples' label according to its support.

Similar to the way it's performed in the original paper, the GNN model pre-trained with MolCLR framework is finetuned. A randomly initialised prediction head with decreasing hidden size replaces the MLP used during pretraining, and binary classification task is performed by considering a graph classification approach. Given a set of attributed graphs

$\mathcal{G} = \{(G_1, y_1), (G_2, y_2),..., (G_n, y_n)\}$, with $\mathcal{Y} = \{y_1, y_2,..., y_n\}$ the set of labels, graph classification is defined as learning a function $f : \mathcal{G} \rightarrow \mathcal{Y}$, with $\mathcal{G}$ the input space of graphs. This is achieved by aggregating the nodes features with a mean pooling layer, i.e. the graph embedding are computed by averaging node features. Models' hyperparameters tuning is performed by considering the same hyperparameters' values used in the MolCLR paper. The model block diagram of the finetuned MolCLR model is reported in Figure 5.4b. The model is trained with cross entropy loss, weighted by class support to account for the datasets class imbalance and optimized with Adam optimizer.

5.2.4 Augmenting QSAR toxicity predictions with semantic knowledge

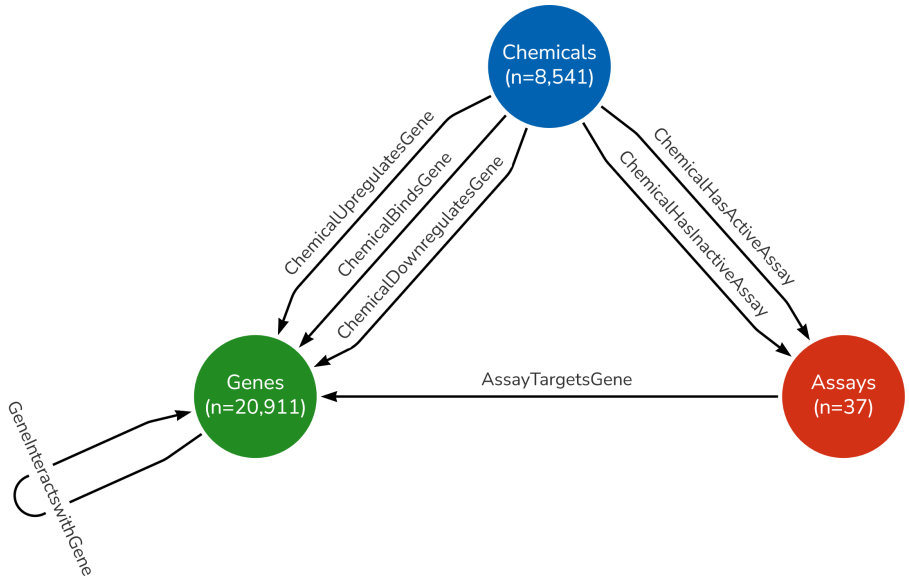


Figure 5.2: Heterogeneous graph used for the Semantic GNN predictive model, comprising chemicals, assays and genes nodes with their cardinality, and the semantic relations between them.

A directed heterogeneous graph $G_s = (\mathcal{C}, \mathcal{T}, \mathcal{K}, \mathcal{R}_s)$ is created to inject the toxicity prediction task with semantic information. Specifically, first the set $\mathcal{T}$ of assay nodes corresponding to the toxicologic assay selected during pre-processing is gathered. Second, the set of chemical nodes $\mathcal{C}$ is defined as all the chemical substances with known activity values in at least one of the selected toxicologic assay of interest, resulting in n = 8,541 unique chemical substances. Finally, the set of gene nodes $\mathcal{K}$ is defined as the ComptoxAI-derived gene entities that are linked to the chemical nodes in $\mathcal{C}$, obtaining a set of n = 20,911 genes (Figure 5.1D). The set of semantic relations $\mathcal{R}_s$ is defined as one of the following: edges between elements

of $\mathcal{T}$ and $\mathcal{C}$, which define whether the chemical is active or inactive with respect to the assay of interest; edges between elements of $\mathcal{C}$ and $\mathcal{K}$, which define either whether a chemical can alter gene expression or that the chemical directly binds to the gene's protein product; edges between two elements of $\mathcal{K}$, indicating gene-gene interactions; and edges between elements of $\mathcal{T}$ and $\mathcal{K}$, indicating the key gene involved in a toxic endpoint. The corresponding graph schema is shown in Figure 5.2.

Hyperparameter name	**Values**
GAT hidden size	[64, 128]
Number of GAT layer	[1, 2]
Dropout ratio	[0, 0.3]
Learning rate	[0.01, 0.001]
Weight decay	[0.01, 0.001, 0.0001]
Layer normalization	[True, False]
Aggregation mode	[sum, mean]

Table 5.4: Semantic GNN model's hyperparameters tuned with their values optimized during grid search.

Similar to Romano et al. [180], the QSAR predictions with semantic knowledge is modelised as a node classification task applied over the chemical nodes, defined as learning a function $f : X_c \in \mathcal{R}^{N_c \times F} \rightarrow \mathcal{Y}$, with X_c a matrix in which each row is a chemical and each column is a numerical vector with dimension F, but differently from the previous work the 2D graph embedding of the molecule is used instead of MACCS fingerprint. The molecules embedding are extracted from the GNN pre-trained module, and used as the initial chemical node representations. Differently, a uniformly sample scalar value is assigned to the assay and gene nodes' features. To avoid information leakage, when considering a specific toxicologic assay as target label, this node and all the edges connecting this node from the semantic graph are deleted.

The semantic model is composed by one or more GAT layer (Eq. 2.29) for each semantic relations, meaning that a parameter-specific message passing step is computed over each couple of nodes but differing according to the edge that connect the nodes. During the model training, each node representation in the semantic graph is updated with the update rule in the Eq. 2.29. In addition, the overall semantic module comprises a prediction head, modelised as a linear layer, that is applied only on the chemical nodes involved in the tasks, in order to train the model with the classification loss and obtain toxicity predictions. This model is referred as *Semantic GNN* and its block diagram is presented in Figure 5.4c. The

model's hyperparameters are tuned with a grid search on the values reported in Table 5.4, and the model is trained for 100 epochs with cross entropy loss weighted by class support and optimized with Adam optimizer.

5.2.5 Unified training of the molecule encoder and semantic modules

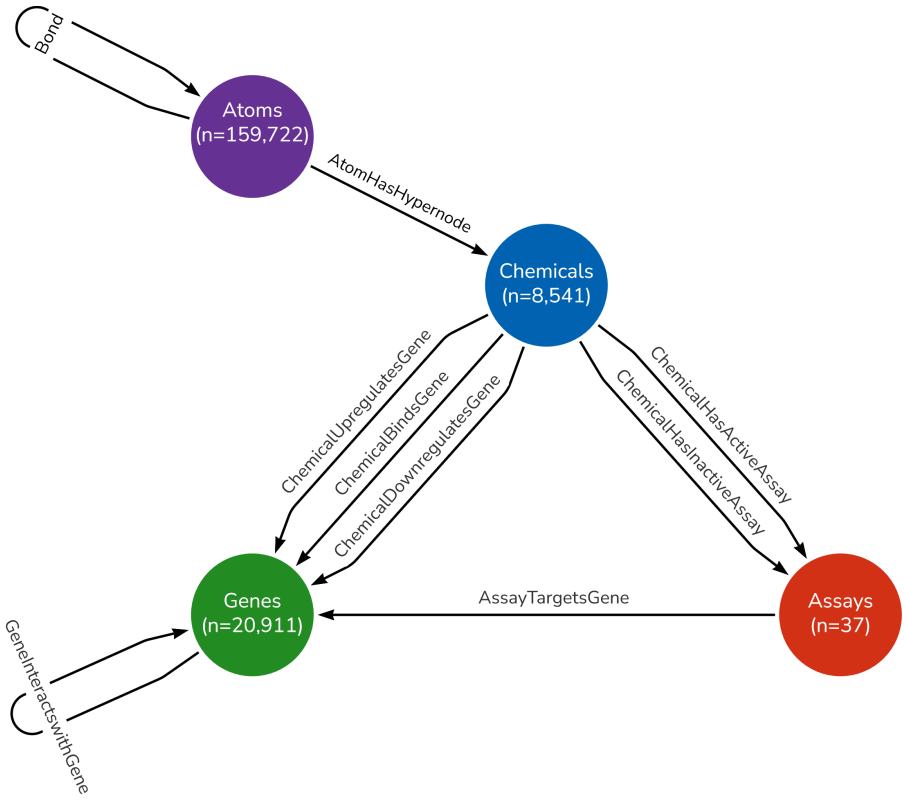
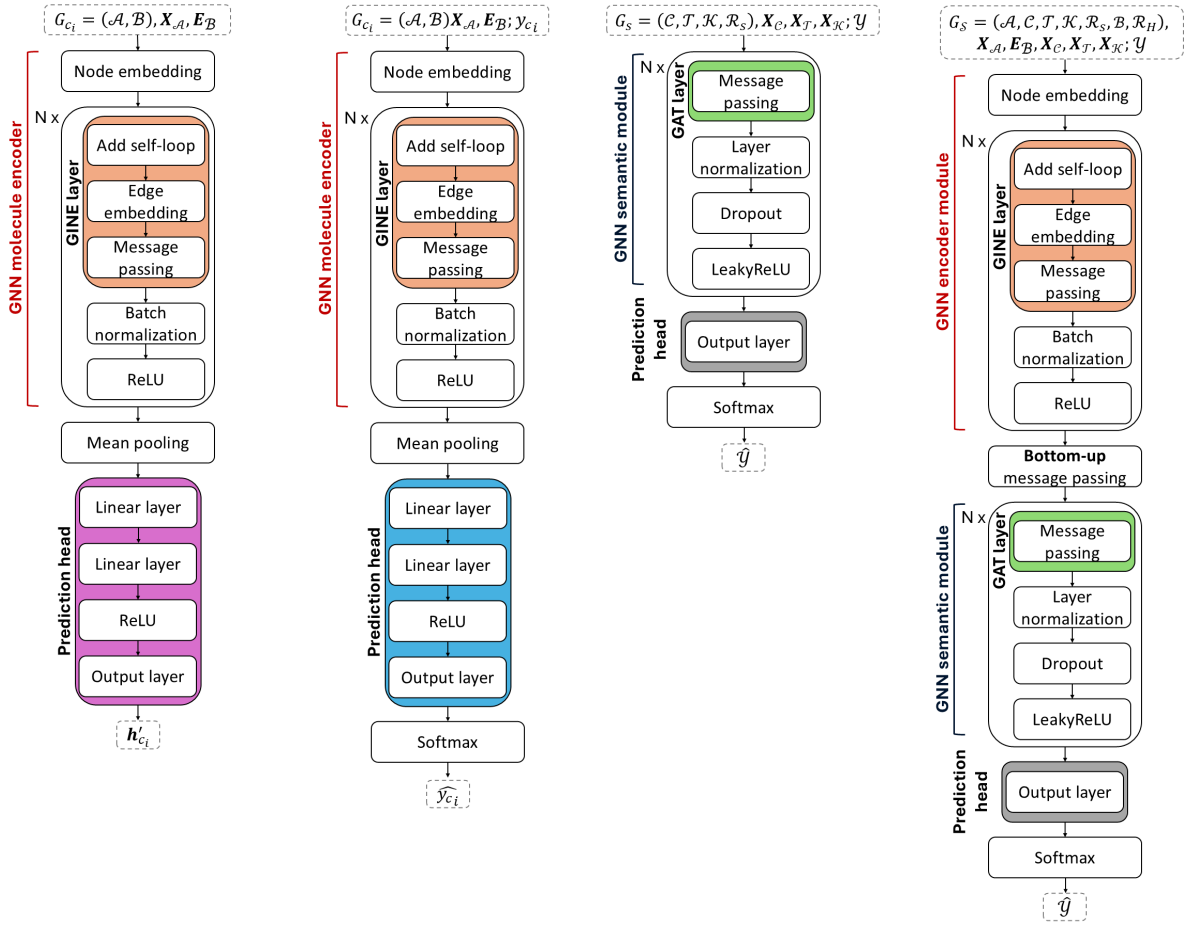


Figure 5.3: Heterogeneous graph used for the MolCLR+Sem predictive model, comprising atoms, chemicals, assays and gene nodes with their cardinality, and the semantic relations between them.

In the semantic module, the molecules embedding are directly used for the downstream prediction task and are updated in a static manner with the semantic knowledge incorporated in Eq. 2.29. Differently, a hierarchical version of the semantic toxicity predictive model is created, in which the molecule embeddings are updated dynamically during the training, meaning that graph embedding of the molecules are not extracted from the molecule encoder but instead this former is trained in combination with the semantic GNN. In order to implement this strategy, the heterogeneous graph (which schema is presented in 5.2) is expanded by representing each chemical node in $\mathcal{C}$ with its 2D graph structure.



(a) MolCLR model for the pretraining of the GNN encoder.

(b) MolCLR model finetuned for obtaining toxicity predictions.

(c) Semantic GNN model trained for toxicity predictions.

(d) MolCLR+Sem model used for toxicity predictions.

Figure 5.4: GNN molecule encoder based on MolCLR, pretrained to learn molecules embedding (a) and finetuned to obtain toxicity predictions (b). Semantic GNN (c) and MolCLR+Sem (d) models, used to inject the toxicity prediction task with semantic knowledge and predict chemicals toxicity.

Specifically, the graph G_s is expanded to include the set $\mathcal{A}$, composed by the atoms that appears in all the molecules considered (n = 159,722), atom features matrix $\mathbf{X}_{\mathcal{A}} \in \mathbb{R}^{N \times 9}$, the set $\mathcal{R}_{\mathcal{B}}$ made by all the edges (bonds) between the atoms and edges features matrix $\mathbf{E} \in \mathbb{R}^{N \times 3}$. By doing this, the resulting graph is $G_s = (\mathcal{C}, \mathcal{T}, \mathcal{K}, \mathcal{R}_s, \mathcal{A}, \mathcal{B}, \mathcal{R}_h)$ became a directed hierarchical graph, in which the chemicals, represented by the graph molecules, lay in the molecule space, while the semantic entities lay in the semantic space. These spaces are connected through the set of edges belonging to $\mathcal{R}_h$, which connect each atom in $\mathcal{A}$ to the respective node chemical in $\mathcal{C}$, making each node chemical a *hypernode*.

Giving this new heterogeneous graph, which schema is showed in Figure 5.3, the GNN

encoder module and the semantic module are combined, and the resulting model is trained for toxicity predictions, referring to it as *MolCLR+Sem* (Figure 5.1E). The unified model is inspired by the work of Zhong et al. [248], in which authors propose a hierarchical formulation of the message passing framework. Giving a graph in which different hierarchical level can be defined, this framework comprises *within-level*, *bottom-up* and *top-down* message propagation, indicating the exchanging of information between nodes within the same level of a graph, from a node that belong to a lower level to a node that belong to a higher level, and from a node that belongs to a higher level to a node that belongs to a lower level, respectively. Since the hierarchical graph is composed of only two levels, the within-level and bottom-up level message propagation are adopted. For the semantic graph, the chemical hypernodes features are initialised with zero vectors, and gene and assay features are initialised with scalar values as used in the previous approach.

The unified model block diagram is reported in Figure 5.4d and comprises the following computations of message passing: i) first within-atom message passing, represented by the modified GIN layer, is applied to update each atom a_i representation to $\mathbf{h}'_{a_i}$ by using the pre-trained GNN module; ii) second, bottom-up propagation is used to compute for each chemical hypernode c_n its representation $\mathbf{h}'_{c_n}$, by averaging the atom features for each atom $a_i \in c_n$. Here, the message flows from atoms in the molecule space (lower level of the hierarchy) to chemical hypernodes in the semantic space (higher level of the hierarchy) and not vice versa; iii) lastly, within-semantic message passing, represented by different GAT layers as the number of semantic relations, is applied to update for each node q in the semantic graph its representation to $\mathbf{h}'_q$, by using the semantic module. Note that hyperparameter tuning is not performed but instead the same values reported in Table 5.4 and resulting from the semantic module tuning are used, and training of the unified model is conducted as in the static approach.

5.2.6 Obtain toxicity predictions with GNNExplainer

GNNExplainer [240] is a post-hoc graph XAI method proposed to explain GNN model predictions. Considering a GNN pre-trained classifier used to obtain embedding for a node v_i through its computational graph $G'_{v_i} \subseteq G$, defined by adjacency matrix $\mathbf{A}'_{v_i}$ and node features matrix $\mathbf{X}'_{v_i}$, by learning a conditional distribution $P(Y|G'_{v_i}, \mathbf{X}'_{v_i})$ indicating the probability of nodes to belong to a specific class, with Y a random variable representing the possible classification labels $\{1,...,L\}$. The objective of GNNExplainer is to obtain an explanation as a mask $\mathbf{M}_{\mathcal{E}}(\xi, f, v_i, y_{v_i}) \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$, in which elements represent important scores associated to the graph edges according to the prediction label y_{v_i}, obtained for a node v_i.

The mask is treated as a trainable parameter matrix and learned through gradient descend optimization of the following cross-entropy objective, between the label class and the model prediction:

$$\min_{\mathbf{M}_{\mathcal{E}}} - \sum_{l=1}^{L} \mathbb{1}[y = l] log P(Y = y | G = \mathbf{A}'_{v_i} \odot \mathbf{M}_{\mathcal{E}}, \mathbf{X} = \mathbf{X}'_v). \tag{5.3}$$

The computational graph G'_{v_i}' and the mask $\mathbf{M}_{\mathcal{E}}$ can be combined through element-wise multiplication $\odot$ to obtain a subgraph $\tilde{G} \subseteq G'_{v_i}$ with adjacency matrix $\tilde{\mathbf{A}}'_{v_i} = \mathbf{A}'_{v_i} \odot \mathbf{M}_{\mathcal{E}}$ that indicates the relevant subsection of the computational graph for the prediction obtained.

It is possible to apply this method to explain why a GNN model takes a decision for a specific input, i.e., study the phenomenon behind the dataset we are observing, or to explain the logic behind the model in choosing a specific output, by computing the gradient regard of the true label or regard the predicted label, respectively [10]. The focus on this works is in using this method to explain the chemicals toxicity for specific toxicologic task.

In particular, given the MolCLR finetuned predictive model, first the edge embeddings are discarded, since not supported by the pyg implementation of GNNExplainer, by simplifying the GNN molecule encoder and using the pre-trained weight as a GIN encoder, and then applying GNNExplainer with input the active chemicals for each toxicologic assay of interest to obtain their relevant substructures. GNNExplainer was trained for 100 epochs with a learning rate value of 0.01 to learn the mask $\mathbf{M}_{\mathcal{E}}$ for each random runs. We then average the mask obtained from each run and apply a threshold to the averaged mask, by keeping only the half of the initial number of edges, ranking them with their importance score.

5.3 Results

5.3.1 Quantitative evaluation of the learned chemicals representations

Quantitative evaluation of the pre-training encoder is conducted on the learned chemicals representations to verify that the embeddings capture meaningful information about chemical functions, by first selecting all chemicals involved in the Tox21 tasks, and then querying ComptoxAI and PubChem to retrieve their chemical and physical properties.

Specifically, chemical properties are encoded by binary variables extracted from MACCS keys, with values 1 or 0, depending if the molecules has or not, respectively, the specific property. The chemical properties considered are: i) aromaticity (MACCS key 162), defined

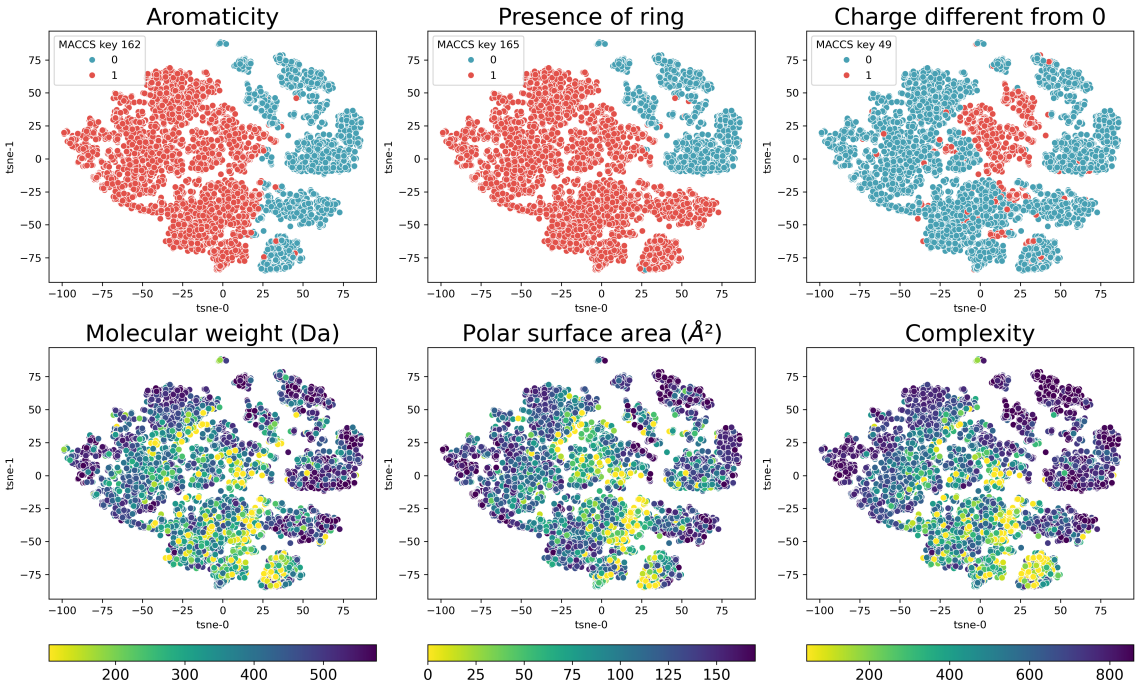


Figure 5.5: Graph embedding computed by the GNN molecule encoder for n = 8541 chemicals, involved in the 37 tox21 tasks. First the embedding are projected in two dimension with t-SNE and then coloured according to chemical (aromaticity, presence of ring and charge different from 0, obtained from MACCS key 162, 165 and 49, respectively) and physical properties (molecular weight, polar surface area and complexity). Considering chemical properties it is possible to observe a clear cluster of aromatic compound on the left, that partially overlap the compounds with a ring, and that the molecule with charge different from 0 tend to stay in the middle of the 2D projections. When considering physical properties, complex and heavy molecules seems to be pushed at the side of the embedding space, while the middle and bottom part are prevalent of smaller molecules.

as the characteristics of molecules to have rings with alternating bonds or lone pairs that are more stable than expected; ii) presence of ring (MACCS key 165), defined as the presence of at least one cycle of atoms and bonds in the molecules; and iii) charge different from 0 (MACCS key 49), where charge is defined as the formal charge assigned to all the atoms within a molecule, calculated based on the distribution of electrons in the chemical bonds.

Differently, physical properties are numerical variables, characterized by a positive value. The physical properties considered are: i) molecular weight measured in Dalton (also called the unified atomic mass unit), identified with symbol *Da*, and expressing the weight of the molecule; ii) polar surface area, measured in angstroms (with symbol $\mathring{A}$ and corresponding to 10^{-10} meters) squared, that is a popular metric in medicinal chemistry since it is used for the optimization of a drug's ability to permeate cells and indicating the surface sum over all polar atoms or molecules; and iii) complexity, that is adimensional and defines a

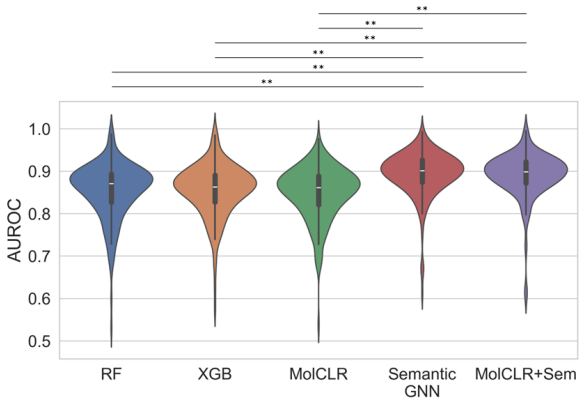
rough estimate of how complicated the structure is (the value extracted from PubChem is computed according to a specific formula [19, 88]).

After embedding are extracted from the pre-trained MolCLR model, these are projected into two dimensions using t-SNE, and coloured according to the values of the selected properties. Results are reported in Figure 5.5. Here it is possible to observe that there is a distinct cluster of aromatic compounds (contained within the larger cluster of all molecules with ring structures), as showed in light-red, as well as a cluster of molecules with non-zero charge, placed at the center of the 2D projected space. Considering the physical properties, less complex and smaller molecules tent to co-organize in the middle of the projected space, moving away from it with increasing radius as the values of these properties increase.

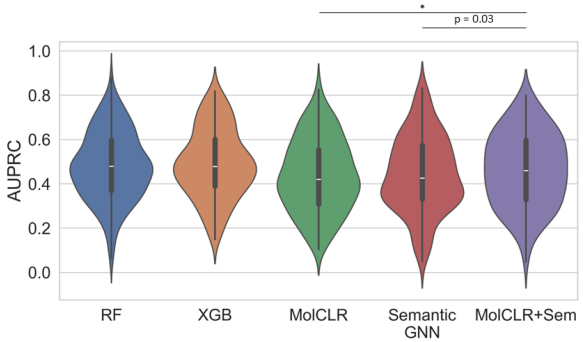
5.3.2 Including semantic knowledge improves toxicity prediction vs. baseline models

Classification and calibration results average on all the Tox21 tasks are reported in Figure 5.6, where p-value is reported only for the comparison between the proposed model (Semantic GNN and MolCLR+Sem), with significance level abbreviated with an asterisk (*) if p-value is < 0.01 and with two asterisks (**) if p-value is < 0.001. As shown in Figure 5.6a, the AUROC is significantly greater (p-value < 0.001 in all pairwise comparisons) when semantic data are included (AUROC = 0.894±0.053 for the Semantic GNN; AUROC = 0.89±0.054 for the MolCLR+Sem), versus the RF (AUROC = 0.845±0.065), XGB (AUROC = 0.855±0.061) and MolCLR (AUROC = 0.849±0.061) baseline models. In addition, the models using semantics shows a lower variance in AUROC scores when compared to the baseline models, suggesting that the semantic strategy is more robust to class imbalance across the assays included in the study. When stratifying on individual assays, in 31 assays either Semantic GNN or MolCLR+Sem attain the highest mean AUROC, as reported in the heatmap in Figure 5.7. Therefore, adding semantic data to QSAR modeling improves toxicity predictions both overall and for individual toxicity assays. In only 4 Tox21 assays, one of the baseline ML models outperforms both the Semantic GNN and MolCLR+Sem models. In addition, there is not a significant difference in AUROC between the Semantic GNN with MolCLR+Sem models.

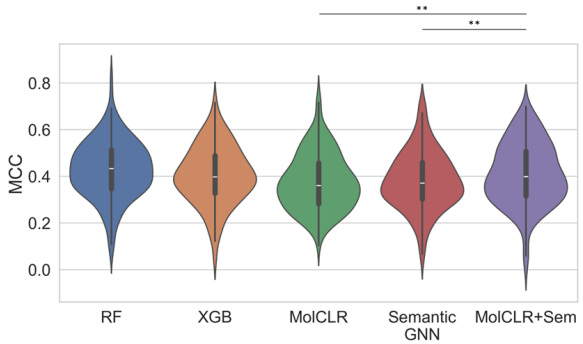
Conversely, when considering model discrimination via AUPRC and MCC, the Semantic GNN yields lower performance when compared with baseline ML models (Figure 5.6bb and Figure 5.6cc). Specifically, the Semantic GNN yields an AUPRC of 0.443±0.171, which is significantly less than the AUPRC for both RF (0.479±0.154; p-value < 0.001), and XGB (0.490±0.155; p-value < 0.001), and has an MCC value of 0.382±0.124, which is



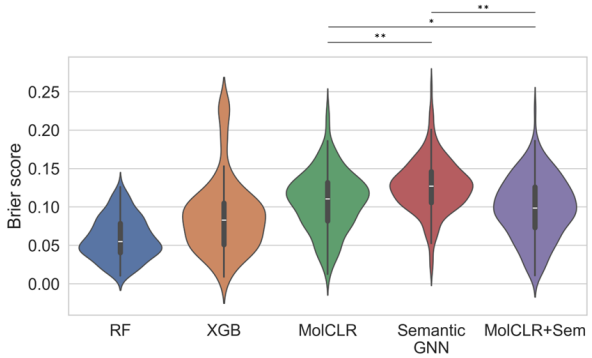
(a) All the pairwise comparisons between Semantic GNN or MolCLR+Sem, and the ML or MolCLR models are statistically significant, showing that AUROC improves when adding knowledge in the toxicity prediction task.



(b) Both MolCLR and Semantic GNN show significant lower values of AUPRC when compared to MolCLR+Sem. In addition, MolCLR model achieve similar performance when compared to the ML models.



(c) Both MolCLR and Semantic GNN show significant lower values of MCC when compared to MolCLR+Sem. In addition, MolCLR model achieve similar performance when compared to the ML models.



(d) Semantic GNN achieves the highest Brier score, and it is outperformed by MolCLR. The predictions' calibration is improved when unifying the training of the two GNN model, i.e., in the MolCLR+Sem model.

Figure 5.6: Classification and calibration performance for the tox21 tasks, computed from 5 random runs for each model. For each metric it is reported the violin plot showing the metric distribution. Area Under the Receiver Operating Characteristic curve - AUROC - (a); Area Under the Precision-Recall Curve - AUPRC (b); Matthews Correlation Coefficient - MCC - (c); Brier score (d). The significance levels are reported as *: p-value < 0.01; **: p-value < 0.001; with p-value resulting from paired t-test and corrected for multiple pairwise comparisons with the Holm-Sidak correction.

also significantly different from that of the RF (0.431 ± 0.123; p-value < 0.001), and XGB (0.405 ± 0.124; p-value $= 0.001$) models. However, the MolCLR+Sem model achieves similar performance to the baseline ML models in terms of AUPRC (0.465 ± 0.166), and MCC (0.409 ± 0.133), which is only significantly different from the RF model (p-value $= 0.012$). Interestingly, MolCLR+Sem shows significantly higher AUPRC and MCC versus MolCLR

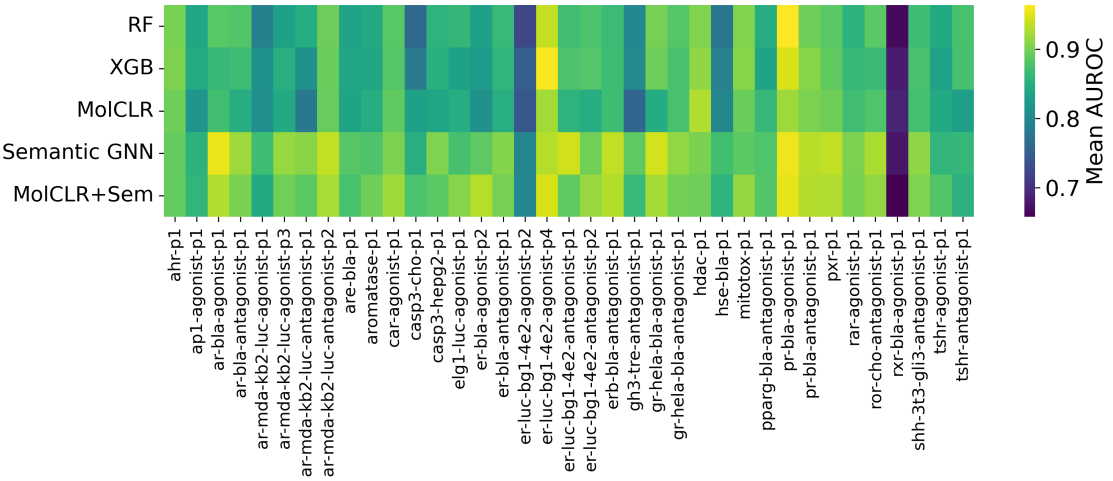


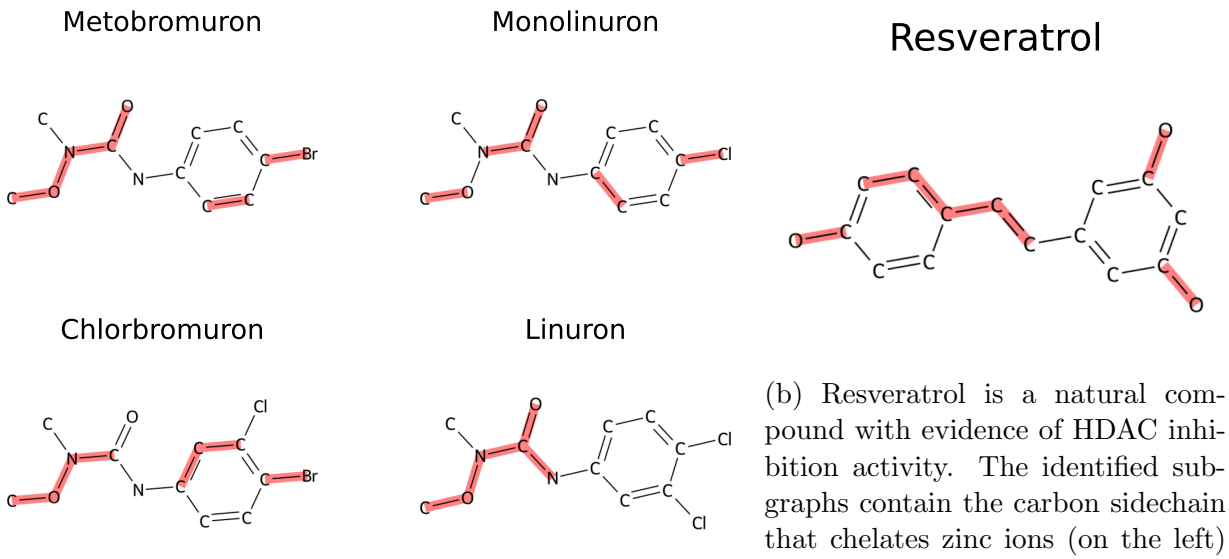
Figure 5.7: Heatmap of mean AUROC values computed from 5 random runs for each model architecture and for each task. Models informed by semantic data generally attain a higher mean AUROC.

(AUPR = 0.437±0.163, MCC = 0.375±0.121) and Semantic GNN. These results suggest that ad-hoc strategies adding semantic data to QSAR models can yield comparable performance when compared to well-established ML models and overcome the limitations encountered by simpler GNN models. Similar conclusions can be made for calibration performance, where MolCLR+Sem yields a significantly lower brier score (0.099±0.04) versus Semantic GNN (0.127±0.034) and MolCLR (0.107±0.039), both with p-value < 0.001 (Figure 5.6dd).

5.3.3 Explaining toxicity predictions: Example using histone deacetylase inhibitors

Activities measured by Tox21 toxicological assays can be grouped into two categories: in the first, chemicals are investigated as potential agonists or antagonists of specific receptors, transcription factors, etc., whose function is related to a specific toxicological endpoint of interest, and thus they act as ligands. This includes the tox21-ap1-agonist-p1 assay (agonism of the transcription factor AP-1). In the second category, chemicals are investigated as activators or inactivators of one or more pathways, where various reactions are used to evaluate the toxicological endpoints of interest (e.g., mitochondrial membrane potential is used to measure the effects of chemicals on mitochondrial function in the tox21-mitotox-p1 assay).

Under the assumption that the optimal ML model for a specific toxicological endpoint of interest detects specific substructures of input chemicals (which in turn are related to the action of the chemical compound and thus to one of the two aforementioned categories of



(a) Four herbicides labelled as active for the selected assay. The relevant subgraph includes the hydroxamic acid functional group. Hydroxamic acids are a well-known class of HDAC inhibitors.

(b) Resveratrol is a natural compound with evidence of HDAC inhibition activity. The identified subgraphs contain the carbon sidechain that chelates zinc ions (on the left) and the sidechain that forms H-bonds and stabilizes the molecule in enzyme binding pockets (on the right).

Figure 5.8: Examples of most relevant substructures for the tox21-hdac-p1 assay obtained with GNNExplainer.

assays), GNNExplainer algorithm is used to explain predictions made for chemicals with regard to the tox21-hdac-p1 assay, which measures inhibition of Histone Deacetylase (HDAC), a class of enzymes that catalyses the removal of acetyl groups from the lysine residue of both histone and nonhistone proteins. HDAC inhibitors interact with HDAC enzymes in a predictable pattern, characterized by binding to the zinc-containing catalytic domain of the HDACs [137]. Groups of HDAC inhibitors are defined according to the zinc binding motif (such as hydroxamic acids, which are characterized by an amino group inserted into a carboxylic acid).

In Figure 5.8a are reported four chemical compounds—all of which are herbicides—that are labelled in the Tox21 dataset as active HDAC inhibitors, and through GNNExplainer is it possible to observe that the most important structural features tend to center around the hydroxamic acid domain. Conversely, Figure 5.8b shows that in the case of Resveratrol (a natural compound with evidence of HDAC inhibition) the most important substructures comprise the sidechains responsible for zinc chelation and hydrogen bonds with the HDAC enzyme [221]. These results are examples of how a graph XAI methods can be used to explain why a chemical compound is defined active, by relying on the GNN predictive model based on the molecular structure.

5.4 Discussion

In **Aim 3** it is shown that the addition of semantic data in QSAR modeling can improve toxicity predictions compared to traditional ML algorithms using well-known structural descriptors. This has been acknowledged previously [180], but in this work GNNs are leveraged to both learn a continuous vector representation of the chemicals of interest from their 2D graph, and to update this representation according to semantic relations mined from CompToxAI.

Since the aforementioned prior work shows that starting with MACCS keys and updating the chemical representation with semantic knowledge improves predictive performance, the focus here is on the application of GNNs, which is a natural evolution considering the simple conversion of molecules to graphs, and that graph databases have become one of the preferred ways to represent complex biomedical knowledge, thanks to their ability to represent multimodal entities as nodes that are connected by semantic relations [121]. As future work, the plan is to incorporate additional structural descriptors in the baseline models [141, 178], and overcome the potential pitfall of the 2D molecule representation with the 3D molecule representation [66]. Specifically, 2D and 3D molecule representation can be combined, by first use a 3D invariant GNN to learn how to capture 3D atomc information and then pass these representation into a 2D GNN which can leverage both the topologies [157].

As part of this study, two ways to inject semantic data into predictive models are investigated, and their performance are evaluated on data sourced from the Tox21 database. Updating pre-trained chemical representations with message passing over the created semantic graph improves discrimination performance, regardless of the percentage of positive samples for each dataset. In addition, when the semantic update is computed in series with the chemicals encoding, the calibration performance improves when comparing with the predictive models using the single GNN modules. This can be explained considering that the inclusion of semantic data alleviates the problem of training GNN predictive models with a big amount of label data and enough positive samples.

Indeed, a limitation of this work is that the models are evaluated on relatively small size datasets and with a low prevalence of positive examples, that is a general obstacle for the field of the computational toxicology [109]. However, compared to previous work, this problem is not faced with over- or under-sampling technique [98], since the interest is in assessing the predictive performance by using the original data. In addition, the prediction performance are evaluated on a bigger number of Tox21 assays, compared to most works that rely only on the Tox21 challenge [130, 99, 96]. Considering that the number of Tox21 assays is growing, new experiments can be conducted as future works by including some of

them, or other chemicals dataset of benchmark that include toxicity outcomes [234].

Lastly, GNNExplainer is used to assess the toxicology predictions' explanations for some active drugs for the tox21-hdac-p1 assay. The results, comprising a compound for which the HDAC inhibition is well-documented and others for which there is no evidence in literature, indicate precise substructure of the molecules, suggesting interesting pathway in the toxicology predictions that can be further investigated to explain association between compound and toxicology endpoint of interest.

However, method such as GNNExplainer, that provides subgraph-level explanations, does not guarantee that the subgraph in output is connected as one fragment. A possible improvement on the interpretation of the XAI results could be in the adoption of specific methods that incorporate pre-existing knowledge on the molecule structure, that is perturbed by leveraging molecular fragmentation, i.e. subgraphs that have a specific chemical function, e.g. carboxylic acid that is an organic acid containing a carboxyl group (-COOH), a carbon atom doubly bonded to an oxygen atom and also bonded to a hydroxyl group (-OH) [235]. In addition, quantitative evaluation of XAI results can be performed by relying on methods that introduce benchmark datasets in which the molecules structures are annotated with ground-truth subgraphs, that are know to influence the chemical property predicted, and suitable metrics [169].

Chapter 6

Conclusions

6.1 Graph Representation Learning to build hybrid AI method

The adoption of a hybrid AI approach for the analysis of biomedical data became essential, considering that biomedical research is driven by expert knowledge, that combines insight derived empirically from existing data with the ones based on theory and experiments. Early AI tools in this field prevalently focus on knowledge representation approach, while the paradigms has shifted in the time to data-driven paradigms such as traditional ML to DL. Nowadays, advanced computational techniques have revolutionized the way knowledge is defined and created in AI, where the information is extracted from a big amount of unstructured data. However, working exclusively by adopting this kind of approach could be at strong risk, since accumulating data without knowledge extraction can led to poor formalization of novel discoveries.

GRL has becoming a transformative technique in medical informatics and bioinformatics and provides a balance between the flexibility of the data-driven approaches and the structured knowledge, since comprises computational model suitable to work on graph. This data structure can efficiently represent biomedical entities, and their connection indicating similarity, semantic or structural associations. The diffusion of GRL paradigms in the biomedical research is favored by the abundance of structured biomedical knowledge, available through KBs that are specialized on different biological domains and can be harmonized to constitute KG. In addition, RL offers a scalable and flexible framework for combining different biomedical data types, ranging from real-world data, to publicly available data repository, by connecting different biological scales.

6.2 Summary of the main findings

In this thesis, GRL AI frameworks that integrate pre-existing knowledge in biomedical data analysis were proposed. In particular, three Aims corresponding to different studies were discussed. The aims deal with a specific biomedical task, using real-world data from a clinical study called INTESTRAT-CAD project, or public available data repository such as Tox21. Each aim has a different way to include the knowledge in the objective of interest, from medical domain knowledge residing in initial phenotypic definition to structured knowledge available in specialized KGs, that depends on the the specific GRL method adopted.

The idea that an initial phenotypic definition, such as the evaluation of a disease severity for a patients' cohort, can be used to guide the computational phenotyping, i.e., the aim of identifying subgroups of patient with similar characteristics, was investigated in **Aim 1**. Here, a semi-supervised TDA-framework called pheTDA was proposed, that first uses the Mapper algorithm to create a topological graph from a dataset of clinical variables, while tuning the Mapper algorithm hyper-parameters by leveraging the initial phenotypic label. Subsequently, from the graph obtained pheTDA discovers communities of nodes by using community detection methods, introducing a new stratification in the patients population, that is characterized with the most discriminative features through the means of ML prediction models, that are trained in a one-vs-rest binary classification setting. In addition, the framework provides assistance while performing computational phenotyping with plots that monitors the graphs statistics related to the Mapper outputs. To fully leverage the Mapper visualization ability, pheTDA can also provide the topological graph enriched with the most discriminative variables values. When applied on a training set deriving from a population of 725 subjects enrolled for the Epifania trial, comprising both CAD patients and control, pheTDA has showed to successfully identify five novel subgroups, one of them only comprised only CAD diabetics patients with a higher CV risk. The trained pheTDA pipeline is applied on the test set for evaluating inference capabilities showing that was able to introduce a more robust partition when compared to spectral clustering, also considering a previous step of dimensionality reduction obtained with UMAP.

The combination of clinical data and RNA-seq variables through the use of a KG and KGE for developing a CAS prediction model was explored in **Aim 2**. In this study, a recently published precision medicine-oriented KG, called PrimeKG, was adopted to combine the different variables available for the patients enrolled in the Epifania trial. First, PrimeKG representation was learned by training and testing different KGE models, selecting RotatE model since obtained the highest KG completion perfomance measured on a test set of the PrimeKG triples. Second, the dataset clinical and gene expression variables were mapped

to the PrimeKG nodes, according to the node types and auxiliary information residing in external knowledge repository or scientific literature. Then, a new knowledge contextualized representation is obtained by defining each patient as a combination of the KGE embeddings corresponding to his/her variables' values. Finally, these representation is given in input to ML classifiers that are trained to predict the CAS severity in three different binary classification task: CAS $\geq 25\%$, CAS $\geq 50\%$ and CAS $\geq 70\%$. In particular, experiments are made by comparing the proposed fusion strategy in both single- and multi-modality settings, by considering separately the clinical and the gene variables, their simple concatenation, and their combination with knowledge. While the results showed that using a representation that combine knowledge with data-driven predictive model improves the classification and calibration perfomance for CAS $\geq 25\%$ and CAS $\geq 50\%$, the clinical variables alone are the best strategy in CAS $\geq 70\%$. This suggests that combining different kind of data for predicting a lower severity of CAS can improve the classification pipeline.

Lastly, in **in Aim 3** was investigated how to augment a toxicity prediction model for small molecules by combining semantic knowledge from ComptoxAI, a KG proposed for computational toxicology. In particular, from this KG was first created a chemicals dataset, where chemicals are represented by their 2D graph structure, with atoms' and edges' attributes encoded as nodes' and edges' features. A GNN encoder was pretrained with MolCLR contrastive learning framework to learn molecules robust representations. The embedding space learned is qualitatively evaluated with dimensionality reduction, showing that molecules were clustered according to chemical and physical properties. Subsequently, information about chemicals, their interaction with genes, and their activity for Tox21 assays, were queried from ComptoxAI and two strategies were created to inject the toxicity prediction task with pre-existing knowledge. In the first an heterogeneous graph was created with chemicals, gene and assay entities, and the chemical features were statically initialised with the embedding extracted from the GNN pretrained encoder. Then a GNN semantic module was used to update the chemical representation with the message from the other entities and update them with knowledge, and finally trained to make toxicity prediction. In the second strategy, the heterogeneous graph was augmented by expanding the chemicals as hypernodes, where each molecule was connected to all the atoms that constitute it. A hierarchical GNN message passing mechanism was used to first update the chemical representation in the molecule space, second to flow this information in the semantic space, constituting the chemical embeddings, subsequently updated with the knowledge from the other entities, and to finally make toxicology predictions. Both the strategies improved the toxicity predictions tasks, with competitive results when compared to classic ML models. In addition, it was showed that by using GNN XAI method it was possible to interpret the results obtained, by

identifying the most important subgraph in the molecule for the toxicity assay considered.

6.3 Future developments and final considerations

The thesis' aims are linked with specific biomedical tasks, i.e., CAD risk stratification, CAS severity and small molecules toxicity prediction. However, the possibilities to extended the the proposed AI frameworks to other applicative fields in different biomedical research are ensured by the flexibility of the GRL methods adopted. The motivation behind the choice of these paradigms was dictated by the GRL ability to balance between the flexibility of data-driven methods, such as clustering algorithm and predictive models, and the structured data representation, that is an advantage for the integration of multiple data types and the medical knowledge.

For example, the semi-supervised TDA framework can be used with different initial phenotypic definition, such as different disease, or by using a phenotypic definition evaluated with an ordinary scale, e.g. a clinical score or guidelines, that measure the likelihood to develop a disease or its initial severity. This approach can be used to redefine patients grouping according to their variables and prior medical knowledge, highlighting the possible differences with the current clinical definition of the disease given by the clinical practice or guidelines. Apart from precision medicine applications, the pipeline proposed in Aim 1 can be generally used as an investigative tool when dealing with high-dimensional and large biomedical dataset. This includes fields where there is more availability of data for their nature, such as molecular biology, or when dealing with biomedical entities, e.g., medical images or genomic sequences, that are embedded with a DL models in a representation space. Considering this last situation, the extension of this framework to inspect the representation learned with graph embedding models in Aim 2 and Aim 3 is straightforward. Indeed, the pipeline can be executed to perform computational phenotyping on the knowledge contextualized patients representation learned with KGE (Aim 2), and to discover potential patterns between chemicals by inspecting the small molecules embeddings also updated with biological knowledge (Aim 3).

The KGE models have already demonstrated their ability in different applicative fields to combine entities of different types for generally predictive task. Even if new powerful encoder are recently published, such as graph-based like GNNs, or even Large Language Models (LLMs)-based, KGE models do not stop to provide a solid methodological background for the creation of model which objective is to inject a dataset of interest with an initial biological understanding. In recently published foundation models, e.g. SHEPPERD and TxGNN, proposed for zero-short rare disease prediction and drug-repurposing prediction, respectively,

KGE are used to give to the AI predictive model based on GNN, an initial knowledge on the particular domain of interest through pretraining on large KG. Nevertheless, the learning technique proposed by KGE models is continually used in this studies, during the training of the predictive model for the downstream task of interest, to avoid catastrophic forgetting that will cause the model to loos the biomedical knowledge pre acquired. For example, considering to expand the analysis in Aim 3 by using the entire ComptoxAI graph or a larger KG, the predictive model proposed for computational toxicology based on GNN can benefit from the solid knowledge representation offered by the KGE when working with heterogeneous graph structure. In addition, could be interesting to investigate how the combination of different biomedical entities is performed by more complex model such as recently KGE foundation models, assessing their capability in practical biomedical domain.

GNNs models are specifically designed to work on graph data and are a natural choice when modeling molecules. The GNN message-passing mechanism offers advantage on shallow embedding approaches, allowing to combine different kinds of data in biomedical research, incorporating structured knowledge, and building an end-to-end prediction model, as reported in Aim 3 for the case of small molecules computational toxicology. Considering to expand the application range from the same domain, predictive pipeline for different chemicals' properties can be constructed, for example for predicting ADME properties of potential drugs, by combining classification and regression tasks in the predictive models. In addition, it is straightforward to extend the pipeline for other applications, such as drug-repurposing. Indeed, in the thesis toxicity prediction was formulated as a node classification task, where node label was defined from the connection to chemicals to assay nodes, and drug-repurposing tasks will simply requires to reformulate it as a link prediction between chemicals and genes entities.

Bibliography

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the Web. *Social Networks*, 25(3):211–230, July 2003.

[2] Monica Agrawal, Marinka Zitnik, and Jure Leskovec. Large-scale analysis of disease pathways in the human interactome. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 23:111–122, 2018.

[3] Saaket Agrawal, Marcus D. R. Klarqvist, Connor Emdin, Aniruddh P. Patel, Manish D. Paranjpe, Patrick T. Ellinor, Anthony Philippakis, Kenney Ng, Puneet Batra, and Amit V. Khera. Selection of 51 predictors from 13,782 candidate multimodal features using machine learning improves coronary artery disease prediction. *Patterns*, 2(12):100364, December 2021.

[4] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48, Rio de Janeiro Brazil, May 2013. ACM.

[5] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Ahmed M. Alaa, Thomas Bolton, Emanuele Di Angelantonio, James H. F. Rudd, and Mihaela van der Schaar. Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK Biobank participants. *PLOS ONE*, 14(5):e0213653, 2019. Publisher: Public Library of Science.

[7] Giuseppe Albi, Alessia Gerbasi, Mattia Chiesa, Gualtiero I. Colombo, Riccardo Bellazzi, and Arianna Dagliati. A Topological Data Analysis Framework for Computa-

tional Phenotyping. In Jose M. Juarez, Mar Marcos, Gregor Stiglic, and Allan Tucker, editors, *Artificial Intelligence in Medicine*, pages 323–327, Cham, 2023. Springer Nature Switzerland.

[8] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82):1–6, 2021.

[9] Emily Alsentzer, Michelle M. Li, Shilpa N. Kobren, Ayush Noori, Undiagnosed Diseases Network, Isaac S. Kohane, and Marinka Zitnik. Few shot learning for phenotype-driven diagnosis of patients with rare genetic diseases, December 2024. Pages: 2022.12.07.22283238.

[10] Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. Graphframex: Towards systematic evaluation of explainability methods for graph neural networks. In *The First Learning on Graphs Conference*, 2022.

[11] Joanna S. Amberger, Carol A. Bocchini, François Schiettecatte, Alan F. Scott, and Ada Hamosh. OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic Acids Research*, 43(Database issue):D789–798, January 2015.

[12] Gøran Troseth Andersen, Aleksandr Ianevski, Mathilde Resell, Naris Pojskic, Hanne-Line Rabben, Synne Geithus, Yosuke Kodama, Tomita Hiroyuki, Denis Kainov, Jon Erik Grønbech, Yoku Hayakawa, Timothy C. Wang, Chun-Mei Zhao, and Duan Chen. Multi-bioinformatics revealed potential biomarkers and repurposed drugs for gastric adenocarcinoma-related gastric intestinal metaplasia. *npj Systems Biology and Applications*, 10(1):1–13, November 2024. Publisher: Nature Publishing Group.

[13] Daniele Andreini, Eleonora Melotti, Chiara Vavassori, Mattia Chiesa, Luca Piacentini, Edoardo Conte, Saima Mushtaq, Martina Manzoni, Eleonora Cipriani, Paolo M. Ravagnani, Antonio L. Bartorelli, and Gualtiero I. Colombo. Whole-Blood Transcriptional Profiles Enable Early Prediction of the Presence of Coronary Atherosclerosis and High-Risk Plaque Features at Coronary CT Angiography. *Biomedicines*, 10(6):1309, June 2022.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors,

3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[15] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016. http://networksciencebook.com.

[16] Christoph Bartenhagen, Hans-Ulrich Klein, Christian Ruckert, Xiaoyi Jiang, and Martin Dugas. Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data. *BMC Bioinformatics*, 11(1):567, November 2010.

[17] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[18] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives, April 2014. arXiv:1206.5538.

[19] Steven H. Bertz. The first general index of molecular complexity. *Journal of the American Chemical Society*, 103(12):3599–3601, 1981. _eprint: https://doi.org/10.1021/ja00402a071.

[20] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008. arXiv:0803.0476 [physics].

[21] Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):D267–D270, January 2004.

[22] Rishi Bommasani, Drew A. Hudson, E. Adeli, R. Altman, Simran Arora, Sydney von Arx, et al. On the Opportunities and Risks of Foundation Models. *ArXiv*, August 2021.

[23] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[24] Kevin W. Boyack, David Newman, Russell J. Duhon, Richard Klavans, Michael Patek, Joseph R. Biberstine, Bob Schijvenaars, André Skupin, Nianli Ma, and Katy Börner. Clustering More than Two Million Biomedical Publications: Comparing the Accuracies of Nine Text-Based Similarity Approaches. *PLOS ONE*, 6(3):e18029, March 2011. Publisher: Public Library of Science.

[25] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[26] W. E. Broadhead, S. H. Gehlbach, F. V. de Gruy, and B. H. Kaplan. The Duke-UNC Functional Social Support Questionnaire. Measurement of social support in family medicine patients. *Medical Care*, 26(7):709–723, July 1988.

[27] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, May 2021. arXiv:2104.13478 [cs].

[28] Hanxuan Cai, Huimin Zhang, Duancheng Zhao, Jingxing Wu, and Ling Wang. FP-GNN: a versatile deep learning architecture for enhanced molecular property prediction. *Briefings in Bioinformatics*, 23(6):bbac408, November 2022.

[29] Michael Caldera, Pisanu Buphamalai, Felix Müller, and Jörg Menche. Interactome-based approaches to human disease. *Current Opinion in Systems Biology*, 3:88–94, June 2017.

[30] Genís Calderer and Marieke L. Kuijjer. Community Detection in Large-Scale Bipartite Biological Networks. *Frontiers in Genetics*, 12, April 2021. Publisher: Frontiers.

[31] T. Caliński and J Harabasz and. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[32] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[33] Grace Casaclang-Verzosa, Sirish Shrestha, Muhammad Jahanzeb Khalil, Jung Sun Cho, Márton Tokodi, Sudarshan Balla, Mohamad Alkhouli, Vinay Badhwar, Jagat Narula, Jordan D. Miller, and Partho P. Sengupta. Network Tomography for Understanding Phenotypic Presentations in Aortic Stenosis. *JACC: Cardiovascular Imaging*, 12(2):236–248, February 2019.

[34] Remzi Celebi, Huseyin Uyar, Erkan Yasar, Ozgur Gumus, Oguz Dikenelli, and Michel Dumontier. Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings. *BMC Bioinformatics*, 20:726, December 2019.

[35] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, February 2023.

[36] David Chang, Ivana Balažević, Carl Allen, Daniel Chawla, Cynthia Brandt, and Richard Andrew Taylor. Benchmark and Best Practices for Biomedical Knowledge Graph Embeddings. *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 2020:167–176, July 2020.

[37] Frédéric Chazal and Bertrand Michel. An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists. *Frontiers in Artificial Intelligence*, 4, September 2021. Publisher: Frontiers.

[38] Guanhua Chen, Xinyue Wang, Qiang Sun, and Zheng-Zheng Tang. Multidimensional scaling improves distance-based clustering for microbiome data. *Bioinformatics*, 41(2):btaf042, February 2025.

[39] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[40] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, November 2020. ISSN: 2640-3498.

[41] Feixiong Cheng, István A. Kovács, and Albert-László Barabási. Network-based prediction of drug combinations. *Nature Communications*, 10(1):1197, March 2019. Publisher: Nature Publishing Group.

[42] Artem Cherkasov, Eugene N. Muratov, Denis Fourches, Alexandre Varnek, Igor I. Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C. Martin, Roberto Todeschini, Viviana Consonni, Victor E. Kuz'min, Richard Cramer, Romualdo Benigni, Chihae Yang, James Rathman, Lothar Terfloth, Johann Gasteiger, Ann Richard, and Alexander Tropsha. QSAR modeling: where have you been? Where are you going to? *Journal of Medicinal Chemistry*, 57(12):4977–5010, June 2014.

[43] Mattia Chiesa, Gualtiero I Colombo, and Luca Piacentini. Damirseq—an r/bioconductor package for data mining of rna-seq data: normalization, feature selection and classification. *Bioinformatics*, 34(8):1416–1418, 12 2017.

[44] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, December 2004. Publisher: American Physical Society.

[45] Cleveland clinic url for coronary artery calcification, 2022. https://my.clevelandclinic.org/health/diseases/22953-coronary-artery-calcification.

[46] Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical Science*, 10(2):370–377, January 2019. Publisher: The Royal Society of Chemistry.

[47] Gennaro Cordasco and Luisa Gargano. Community detection via semi-synchronous label propagation algorithms. In *2010 IEEE International Workshop on: Business Applications of Social Network Analysis (BASNA)*, pages 1–8, December 2010.

[48] Julian Cremer, Leonardo Medrano Sandonas, Alexandre Tkatchenko, Djork-Arné Clevert, and Gianni De Fabritiis. Equivariant Graph Neural Networks for Toxicity Prediction. *Chemical Research in Toxicology*, 36(10):1561–1573, October 2023. Publisher: American Chemical Society.

[49] Ricardo C. Cury, Jonathon Leipsic, Suhny Abbara, Stephan Achenbach, Daniel Berman, Marcio Bittencourt, Matthew Budoff, Kavitha Chinnaiyan, Andrew D. Choi, Brian Ghoshhajra, Jill Jacobs, Lynne Koweek, John Lesser, Christopher Maroules, Geoffrey D. Rubin, Frank J. Rybicki, Leslee J. Shaw, Michelle C. Williams, Eric Williamson, Charles S. White, Todd C. Villines, and Ron Blankstein. CAD-RADS™ 2.0 - 2022 Coronary Artery Disease-Reporting and Data System: An Expert Consensus Document of the Society of Cardiovascular Computed Tomography (SCCT), the American College of Cardiology (ACC), the American College of Radiology (ACR), and the North America Society of Cardiovascular Imaging (NASCI). *Journal of Cardiovascular Computed Tomography*, 16(6):536–557, November 2022. Publisher: Elsevier.

[50] Anant Dadu, Vipul K. Satone, Rachneet Kaur, Mathew J. Koretsky, Hirotaka Iwaki, Yue A. Qi, Daniel M. Ramos, Brian Avants, Jacob Hesterman, Roger Gunn, Mark R. Cookson, Michael E. Ward, Andrew B. Singleton, Roy H. Campbell, Mike A. Nalls, and Faraz Faghri. Application of Aligned-UMAP to longitudinal biomedical studies. *Patterns*, 4(6):100741, June 2023.

[51] Arianna Dagliati, Nophar Geifman, Niels Peek, John H. Holmes, Lucia Sacchi, Riccardo Bellazzi, Seyed Erfan Sajjadi, and Allan Tucker. Using topological data analysis and pseudo time series to infer temporal phenotypes from electronic health records. *Artificial Intelligence in Medicine*, 108:101930, August 2020.

[52] Arianna Dagliati, Zachary H. Strasser, Zahra Shakeri Hossein Abad, Jeffrey G. Klann, Kavishwar B. Wagholikar, Rebecca Mesa, Shyam Visweswaran, Michele Morris, Yuan Luo, Darren W. Henderson, Malarkodi Jebathilagam Samayamuthu, Bryce W.Q. Tan, Guillame Verdy, Gilbert S. Omenn, Zongqi Xia, Riccardo Bellazzi, James R. Aaron, Giuseppe Agapito, Adem Albayrak, Giuseppe Albi, Mario Alessiani, Anna Alloni, Danilo F. Amendola, François Angoulvant, Li L.L.J. Anthony, Bruce J. Aronow, Fatima Ashraf, Andrew Atz, Paul Avillach, Paula S. Azevedo, James Balshi, Brett K. Beaulieu-Jones, Douglas S. Bell, Antonio Bellasi, Riccardo Bellazzi, Vincent Benoit, Michele Beraghi, José Luis Bernal-Sobrino, Mélodie Bernaux, Romain Bey, Surbhi Bhatnagar, Alvar Blanco-Martínez, Clara-Lea Bonzel, John Booth, Silvano Bosari, Florence T. Bourgeois, Robert L. Bradford, Gabriel A. Brat, Stéphane Bréant, Nicholas W. Brown, Raffaele Bruno, William A. Bryant, Mauro Bucalo, Emily Bucholz, Anita Burgun, Tianxi Cai, Mario Cannataro, Aldo Carmona, Charlotte Caucheteux, Julien Champ, Jin Chen, Krista Y. Chen, Luca Chiovato, Lorenzo Chiudinelli, Kelly Cho, James J. Cimino, Tiago K. Colicchio, Sylvie Cormont, Sébastien Cossin, Jean B. Craig, Juan Luis Cruz-Bermúdez, Jaime Cruz-Rojo, Arianna Dagliati, Mohamad Daniar, Christel Daniel, Priyam Das, Batsal Devkota, Audrey Dionne, Rui Duan, Julien Dubiel, Scott L. DuVall, Loic Esteve, Hossein Estiri, Shirley Fan, Robert W. Follett, Thomas Ganslandt, Noelia García Barrio, Lana X. Garmire, Nils Gehlenborg, Emily J. Getzen, Alon Geva, Tobias Gradinger, Alexandre Gramfort, Romain Griffier, Nicolas Griffon, Olivier Grisel, Alba Gutiérrez-Sacristán, Larry Han, David A. Hanauer, Christian Haverkamp, Derek Y. Hazard, Bing He, Darren W. Henderson, Martin Hilka, Yuk-Lam Ho, John H. Holmes, Chuan Hong, Kenneth M. Huling, Meghan R. Hutch, Richard W. Issitt, Anne Sophie Jannot, Vianney Jouhet, Ramakanth Kavuluru, Mark S. Keller, Chris J. Kennedy, Daniel A. Key, Katie Kirchoff, Jeffrey G. Klann, Isaac S. Kohane, Ian D. Krantz, Detlef Kraska, Ashok K. Krishnamurthy, Sehi L'Yi, Trang T. Le, Judith Leblanc, Guillaume Lemaitre, Leslie Lenert, Damien Leprovost, Molei Liu, Ne Hooi Will Loh, Qi Long, Sara Lozano-Zahonero, Yuan Luo, Kristine E. Lynch, Sadiqa Mahmood, Sarah E. Maidlow, Adeline Makoudjou, Alberto Malovini, Kenneth D. Mandl, Chengsheng Mao, Anupama Maram, Patricia Martel, Marcelo R. Martins, Jayson S. Marwaha, Aaron J. Masino, Maria Mazzitelli, Arthur Mensch, Marianna Milano, Marcos F. Minicucci, Bertrand Moal, Taha Mohseni

Ahooyi, Jason H. Moore, Cinta Moraleda, Jeffrey S. Morris, Michele Morris, Karyn L. Moshal, Sajad Mousavi, Danielle L. Mowery, Douglas A. Murad, Shawn N. Murphy, Thomas P. Naughton, Carlos Tadeu Breda Neto, Antoine Neuraz, Jane Newburger, Kee Yuan Ngiam, Wanjiku F.M. Njoroge, James B. Norman, Jihad Obeid, Marina P. Okoshi, Karen L. Olson, Gilbert S. Omenn, Nina Orlova, Brian D. Ostasiewski, Nathan P. Palmer, Nicolas Paris, Lav P. Patel, Miguel Pedrera-Jiménez, Emily R. Pfaff, Ashley C. Pfaff, Danielle Pillion, Sara Pizzimenti, Hans U. Prokosch, Robson A. Prudente, Andrea Prunotto, Víctor Quirós-González, Rachel B. Ramoni, Maryna Raskin, Siegbert Rieg, Gustavo Roig-Domínguez, Pablo Rojo, Paula Rubio-Mayo, Paolo Sacchi, Carlos Sáez, Elisa Salamanca, Malarkodi Jebathilagam Samayamuthu, L. Nelson Sanchez-Pinto, Arnaud Sandrin, Nandhini Santhanam, Janaina C.C. Santos, Fernando J. Sanz Vidorreta, Maria Savino, Emily R. Schriver, Petra Schubert, Juergen Schuettler, Luigia Scudeller, Neil J. Sebire, Pablo Serrano-Balazote, Patricia Serre, Arnaud Serret-Larmande, Mohsin Shah, Zahra Shakeri Hossein Abad, Domenick Silvio, Piotr Sliz, Jiyeon Son, Charles Sonday, Andrew M. South, Anastasia Spiridou, Zachary H. Strasser, Amelia L.M. Tan, Bryce W.Q. Tan, Byorn W.L. Tan, Suzana E. Tanni, Deanne M. Taylor, Ana I. Terriza-Torres, Valentina Tibollo, Patric Tippmann, Emma M.S. Toh, Carlo Torti, Enrico M. Trecarichi, Yi-Ju Tseng, Andrew K. Vallejos, Gael Varoquaux, Margaret E. Vella, Guillaume Verdy, Jill-Jênn Vie, Shyam Visweswaran, Michele Vitacca, Kavishwar B. Wagholikar, Lemuel R. Waitman, Xuan Wang, Demian Wassermann, Griffin M. Weber, Martin Wolkewitz, Scott Wong, Zongqi Xia, Xin Xiong, Ye Ye, Nadir Yehya, William Yuan, Alberto Zambelli, Harrison G. Zhang, Daniela Zo¨ller, Valentina Zuccaro, Chiara Zucco, Shawn N. Murphy, John H. Holmes, and Hossein Estiri. Characterization of long COVID temporal subphenotypes by distributed representation learning from electronic health record data: a cohort study. *eClinicalMedicine*, 64:102210, October 2023.

[53] Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. The Comparative Toxicogenomics Database: update 2019. *Nucleic Acids Research*, 47(D1):D948–D954, January 2019.

[54] Kevin Dawson, Raymond L. Rodriguez, and Wasyl Malyj. Sample phenotype clusters in high-density oligonucleotide microarray data sets are revealed using Isomap, a nonlinear algorithm. *BMC Bioinformatics*, 6(1):195, August 2005.

[55] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. *Proceedings of the AAAI Conference on*

Artificial Intelligence, 32(1), April 2018. Number: 1.

[56] Alex Diaz-Papkovich, Luke Anderson-Trocmé, Chief Ben-Eghan, and Simon Gravel. UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. *PLOS Genetics*, 15(11):e1008432, November 2019.

[57] Joseph A. DiMasi, Henry G. Grabowski, and Ronald W. Hansen. Innovation in the pharmaceutical industry: New estimates of rd costs. *Journal of Health Economics*, 47:20–33, 2016.

[58] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 135–144, New York, NY, USA, 2017. Association for Computing Machinery.

[59] Arkadiusz Z. Dudek, Tomasz Arodz, and Jorge Gálvez. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Combinatorial Chemistry & High Throughput Screening*, 9(3):213–228, March 2006.

[60] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, November 2002. Publisher: American Chemical Society.

[61] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, November 2002. Publisher: American Chemical Society.

[62] Yasha Ektefaie, George Dasoulas, Ayush Noori, Maha Farhat, and Marinka Zitnik. Multimodal learning with graphs. *Nature Machine Intelligence*, 5(4):340–350, April 2023. Publisher: Nature Publishing Group.

[63] Peter Ertl, Eva Altmann, and Jeffrey M. McKenna. The Most Common Functional Groups in Bioactive Molecules and How Their Popularity Has Evolved over Time. *Journal of Medicinal Chemistry*, 63(15):8408–8418, August 2020. Publisher: American Chemical Society.

[64] M. Ester, H. P . Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231. AAAI Press, 1996.

[65] Faraz Faghri, Fabian Brunn, Anant Dadu, Adriano Chiò, Andrea Calvo, Cristina Moglia, Antonio Canosa, Umberto Manera, Rosario Vasta, Francesca Palumbo, Alessandro Bombaci, Maurizio Grassano, Maura Brunetti, Federico Casale, Giuseppe Fuda, Paolina Salamone, Barbara Iazzolino, Laura Peotta, Paolo Cugnasco, Giovanni De Marco, Maria Claudia Torrieri, Salvatore Gallone, Marco Barberis, Luca Sbaiz, Salvatore Gentile, Alessandro Mauro, Letizia Mazzini, Fabiola De Marchi, Lucia Corrado, Sandra D'Alfonso, Antonio Bertolotto, Daniele Imperiale, Marco De Mattei, Salvatore Amarù, Cristoforo Comi, Carmelo Labate, Fabio Poglio, Luigi Ruiz, Lucia Testa, Eugenia Rota, Paolo Ghiglione, Nicola Launaro, Alessia Di Sapio, Jessica Mandrioli, Nicola Fini, Ilaria Martinelli, Elisabetta Zucchi, Giulia Gianferrari, Cecilia Simonini, Stefano Meletti, Rocco Liguori, Veria Vacchiano, Fabrizio Salvi, Ilaria Bartolomei, Roberto Michelucci, Pietro Cortelli, Rita Rinaldi, Anna Maria Borghi, Andrea Zini, Elisabetta Sette, Valeria Tugnoli, Maura Pugliatti, Elena Canali, Luca Codeluppi, Franco Valzania, Lucia Zinno, Giovanni Pavesi, Doriana Medici, Giovanna Pilurzi, Emilio Terlizzi, Donata Guidetti, Silvia De Pasqua, Mario Santangelo, Patrizia De Massis, Martina Bracaglia, Mario Casmiro, Pietro Querzani, Simonetta Morresi, Marco Longoni, Alberto Patuelli, Susanna Malagù, Marco Currò Dossi, Simone Vidale, Salvatore Ferro, Elisabetta Zucchi, Ilaria Martinelli, Letizia Mazzini, Rosario Vasta, Antonio Canosa, Cristina Moglia, Andrea Calvo, Michael A Nalls, Roy H Campbell, Jessica Mandrioli, Bryan J Traynor, and Adriano Chiò. Identifying and predicting amyotrophic lateral sclerosis clinical subgroups: a population-based machine-learning study. *The Lancet Digital Health*, 4(5):e359–e369, May 2022.

[66] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, February 2022. Publisher: Nature Publishing Group.

[67] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, October 2016.

[68] Matthias Fey and J. E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds., March 2019.

[69] Padraig Fitzpatrick, Anna Jurek-Loughrey, Paweł Dłotko, and Jesus Martinez Del Rincon. Ensemble learning for mapper parameter optimization. In *2023 IEEE 35th*

International Conference on Tools with Artificial Intelligence (ICTAI), pages 129–134, 2023.

[70] Mikhail Galkin, Etienne G. Denis, Jiapeng Wu, and William L. Hamilton. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[71] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *ICLR*. OpenReview.net, 2024.

[72] Giuseppina Gallucci, Alfredo Tartarone, Rosa Lerose, Anna Vittoria Lalinga, and Alba Maria Capobianco. Cardiovascular risk of smoking and benefits of smoking cessation. *Journal of Thoracic Disease*, 12(7):3866–3876, July 2020.

[73] Javier O. Garcia, Arian Ashourvan, Sarah Muldoon, Jean M. Vettel, and Danielle S. Bassett. Applications of Community Detection Techniques to Brain Graphs: Algorithmic Considerations and Implications for Neural Function. *Proceedings of the IEEE*, 106(5):846–867, May 2018.

[74] Michael A Gargano, Nicolas Matentzoglu, Ben Coleman, Eunice B Addo-Lartey, Anna V Anagnostopoulos, Joel Anderton, Paul Avillach, Anita M Bagley, Eduard Bakštein, James P Balhoff, Gareth Baynam, Susan M Bello, Michael Berk, Holli Bertram, Somer Bishop, Hannah Blau, David F Bodenstein, Pablo Botas, Kaan Boztug, Jolana Čady, Tiffany J Callahan, Rhiannon Cameron, Seth J Carbon, Francisco Castellanos, J Harry Caufield, Lauren E Chan, Christopher G Chute, Jaime Cruz-Rojo, Noémi Dahan-Oliel, Jon R Davids, Maud de Dieuleveult, Vinicius de Souza, Bert B A de Vries, Esther de Vries, J Raymond DePaulo, Beata Derfalvi, Ferdinand Dhombres, Claudia Diaz-Byrd, Alexander J M Dingemans, Bruno Donadille, Michael Duyzend, Reem Elfeky, Shahim Essaid, Carolina Fabrizzi, Giovanna Fico, Helen V Firth, Yun Freudenberg-Hua, Janice M Fullerton, Davera L Gabriel, Kimberly Gilmour, Jessica Giordano, Fernando S Goes, Rachel Gore Moses, Ian Green, Matthias Griese, Tudor Groza, Weihong Gu, Julia Guthrie, Benjamin Gyori, Ada Hamosh, Marc Hanauer, Kateřina Hanušová, Yongqun (Oliver) He, Harshad Hegde, Ingo Helbig, Kateřina Holasová, Charles Tapley Hoyt, Shangzhi Huang, Eric Hurwitz, Julius O B Jacobsen, Xiaofeng Jiang, Lisa Joseph, Kamyar Keramatian, Bryan King, Katrin Knoflach, David A Koolen, Megan L Kraus, Carlo Kroll, Maaike Kusters, Markus S Ladewig, David

Lagorce, Meng-Chuan Lai, Pablo Lapunzina, Bryan Laraway, David Lewis-Smith, Xiarong Li, Caterina Lucano, Marzieh Majd, Mary L Marazita, Victor Martinez-Glez, Toby H McHenry, Melvin G McInnis, Julie A McMurry, Michaela Mihulová, Caitlin E Millett, Philip B Mitchell, Veronika Moslerová, Kenji Narutomi, Shahrzad Nematollahi, Julian Nevado, Andrew A Nierenberg, Nikola Novák Čajbiková, Jr. Nurnberger, John I, Soichi Ogishima, Daniel Olson, Abigail Ortiz, Harry Pachajoa, Guiomar Perez de Nanclares, Amy Peters, Tim Putman, Christina K Rapp, Ana Rath, Justin Reese, Lauren Rekerle, Angharad M Roberts, Suzy Roy, Stephan J Sanders, Catharina Schuetz, Eva C Schulte, Thomas G Schulze, Martin Schwarz, Katie Scott, Dominik Seelow, Berthold Seitz, Yiping Shen, Morgan N Similuk, Eric S Simon, Balwinder Singh, Damian Smedley, Cynthia L Smith, Jake T Smolinsky, Sarah Sperry, Elizabeth Stafford, Ray Stefancsik, Robin Steinhaus, Rebecca Strawbridge, Jagadish Chandrabose Sundaramurthi, Polina Talapova, Jair A Tenorio Castano, Pavel Tesner, Rhys H Thomas, Audrey Thurm, Marek Turnovec, Marielle E van Gijn, Nicole A Vasilevsky, Markéta Vlčková, Anita Walden, Kai Wang, Ron Wapner, James S Ware, Addo A Wiafe, Samuel A Wiafe, Lisa D Wiggins, Andrew E Williams, Chen Wu, Margot J Wyrwoll, Hui Xiong, Nefize Yalin, Yasunori Yamamoto, Lakshmi N Yatham, Anastasia K Yocum, Allan H Young, Zafer Yüksel, Peter P Zandi, Andreas Zankl, Ignacio Zarante, Miroslav Zvolský, Sabrina Toro, Leigh C Carmody, Nomi L Harris, Monica C Munoz-Torres, Daniel Danis, Christopher J Mungall, Sebastian Köhler, Melissa A Haendel, and Peter N Robinson. The human phenotype ontology in 2024: phenotypes around the world. *Nucleic Acids Research*, 52(D1):D1333–D1346, 11 2023.

[75] Alexander J. Gates, Ian B. Wood, William P. Hetrick, and Yong-Yeol Ahn. Element-centric clustering comparison unifies overlaps and hierarchy. *Scientific Reports*, 9(1):8574, June 2019. Publisher: Nature Publishing Group.

[76] Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B R Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L Blundell, Michael M Bronstein, and Jake P Taylor-King. Utilizing graph machine learning within drug discovery and development. *Briefings in Bioinformatics*, 22(6):bbab159, November 2021.

[77] Yin Ge and Thomas J. Wang. Circulating, Imaging, and Genetic Biomarkers in Cardiovascular Risk Prediction. *Trends in Cardiovascular Medicine*, 21(4):105–112, May 2011.

[78] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1263–1272. PMLR, July 2017. ISSN: 2640-3498.

[79] Kwang-Il Goh, Michael E. Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, May 2007. Publisher: Proceedings of the National Academy of Sciences.

[80] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[81] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, 1971.

[82] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA, 2016. Association for Computing Machinery.

[83] Christopher M. Grulke, Antony J. Williams, Inthirany Thillanadarajah, and Ann M. Richard. EPA's DSSTox database: History of development of a curated chemistry resource supporting computational toxicology research. *Computational Toxicology*, 12:100096, November 2019.

[84] Emre Guney, Jörg Menche, Marc Vidal, and Albert-László Barábasi. Network-based in silico drug efficacy screening. *Nature Communications*, 7(1):10331, February 2016. Publisher: Nature Publishing Group.

[85] Mengfei Guo, Yanan Yu, Tiancai Wen, Xiaoping Zhang, Baoyan Liu, Jin Zhang, Runshun Zhang, Yanning Zhang, and Xuezhong Zhou. Analysis of disease comorbidity patterns in a large-scale China population. *BMC Medical Genomics*, 12(12):177, December 2019.

[86] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[87] William L Hamilton, Rex Ying, and Jure Leskovec. Representation Learning on Graphs: Methods and Applications, April 2018. arXiv.1709.05584.

[88] James B. Hendrickson, Ping Huang, and A. Glenn Toczko. Molecular complexity: a simplified formula adapted to individual atoms. *Journal of Chemical Information and Computer Sciences*, 27(2):63–67, 1987. _eprint: https://doi.org/10.1021/ci00054a004.

[89] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L. Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E. Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing, September 2017. Publisher: eLife Sciences Publications Limited.

[90] Timothy S. C. Hinks, Xiaoying Zhou, Karl J. Staples, Borislav D. Dimitrov, Alexander Manta, Tanya Petrossian, Pek Y. Lum, Caroline G. Smith, Jon A. Ward, Peter H. Howarth, Andrew F. Walls, Stephan D. Gadola, and Ratko Djukanović. Innate and adaptive T cells in asthmatic patients: Relationship to severity and disease mechanisms. *Journal of Allergy and Clinical Immunology*, 136(2):323–333, August 2015. Publisher: Elsevier.

[91] Geoffrey E Hinton and Sam T Roweis. Stochastic Neighbor Embedding. In *Neural Information Processing Systems*, pages 833–840, 2002.

[92] John H. Holmes. *Artificial Intelligence*, pages 221–230. Springer International Publishing, Cham, 2023.

[93] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989.

[94] Weihua Hu, Bowen Liu, Joseph Gomes, M. Zitnik, Percy Liang, V. Pande, and J. Leskovec. Strategies for Pre-training Graph Neural Networks. In *Proceedings of the 8th International Conference on Learning Representation (ICLR)*, 2020.

[95] Kexin Huang, Payal Chandak, Qianwen Wang, Shreyas Havaldar, Akhil Vaid, Jure Leskovec, Girish N. Nadkarni, Benjamin S. Glicksberg, Nils Gehlenborg, and Marinka Zitnik. A foundation model for clinician-centered drug repurposing. *Nature Medicine*, pages 1–13, September 2024. Publisher: Nature Publishing Group.

[96] Ruili Huang, Menghang Xia, Dac-Trung Nguyen, Tongan Zhao, Srilatha Sakamuru, Jinghua Zhao, Sampada A. Shahane, Anna Rossoshek, and Anton Simeonov. Tox21Challenge to Build Predictive Models of Nuclear Receptor and Stress Response

Pathways as Mediated by Exposure to Environmental Chemicals and Drugs. *Frontiers in Environmental Science*, 3, January 2016. Publisher: Frontiers.

[97] Nathan C. Hurley, Adrian D. Haimovich, R. Andrew Taylor, and Bobak J. Mortazavi. Visualization of emergency department clinical data for interpretable patient phenotyping. *Smart Health*, 25:100285, September 2022.

[98] Gabriel Idakwo, Sundar Thangapandian, Joseph Luttrell, Yan Li, Nan Wang, Zhaoxian Zhou, Huixiao Hong, Bei Yang, Chaoyang Zhang, and Ping Gong. Structure–activity relationship-based chemical classification of highly imbalanced Tox21 datasets. *Journal of Cheminformatics*, 12(1):66, October 2020.

[99] Gabriel Idakwo, Sundar Thangapandian, Joseph Luttrell, Zhaoxian Zhou, Chaoyang Zhang, and Ping Gong. Deep Learning-Based Structure-Activity Relationship Modeling for Multi-Category Toxicity Classification: A Case Study of 10K Tox21 Chemicals With High-Throughput Cell-Based Androgen Receptor Bioassay Data. *Frontiers in Physiology*, 10, August 2019. Publisher: Frontiers.

[100] Ryo Ishibashi. Multidimensional scaling methods can reconstruct genomic DNA loops using Hi-C data properties. *PLOS ONE*, 18(8):e0289651, 2023. Publisher: Public Library of Science.

[101] Gábor Iván and Vince Grolmusz. When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks. *Bioinformatics*, 27(3):405–407, February 2011.

[102] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, February 2022. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[103] Roby Joehanes, Andrew D. Johnson, Jennifer J. Barb, Nalini Raghavachari, Poching Liu, Kimberly A. Woodhouse, Christopher J. O'Donnell, Peter J. Munson, and Daniel Levy. Gene expression analysis of whole blood, peripheral blood mononuclear cells, and lymphoblastoid cell lines from the Framingham Heart Study. *Physiological Genomics*, 44(1):59–75, January 2012.

[104] Ruth Johnson, Michelle M. Li, Ayush Noori, Owen Queen, and Marinka Zitnik. Graph Artificial Intelligence in Medicine. *Annual Review of Biomedical Data Science*, 7(1):345–368, August 2024.

[105] Minoru Kanehisa, Miho Furumichi, Yoko Sato, Yuriko Matsuura, and Mari Ishiguro-Watanabe. KEGG: biological systems database as a model of the real world. *Nucleic Acids Research*, 53(D1):D672–D677, January 2025.

[106] William B. Kannel and Ramachandran S. Vasan. Adverse Consequences of the 50% Misconception. *American Journal of Cardiology*, 103(3):426–427, February 2009. Publisher: Elsevier.

[107] Baris Kayalibay, Grady Jensen, and Patrick van der Smagt. CNN-based Segmentation of Medical Imaging Data, July 2017. arXiv:1701.03056 [cs].

[108] Umesh N. Khot, Monica B. Khot, Christopher T. Bajzer, Shelly K. Sapp, E. Magnus Ohman, Sorin J. Brener, Stephen G. Ellis, A. Michael Lincoff, and Eric J. Topol. Prevalence of conventional risk factors in patients with coronary heart disease. *JAMA*, 290(7):898–904, 08 2003.

[109] Changhun Kim, Jaeseong Jeong, and Jinhee Choi. Effects of Class Imbalance and Data Scarcity on the Performance of Binary Classification Machine Learning Models Developed Based on ToxCast/Tox21 Assay Data. *Chemical Research in Toxicology*, 35(12):2219–2226, December 2022. Publisher: American Chemical Society.

[110] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, January 2023.

[111] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations (ICLR), Poster., December 2014.

[112] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[113] Craig Knox, Mike Wilson, Christen M. Klinger, Mark Franklin, Eponine Oler, Alex Wilson, Allison Pon, Jordan Cox, Na Eun Lucy Chin, Seth A. Strawbridge, Marysol Garcia-Patino, Ray Kruger, Aadhavya Sivakumaran, Selena Sanford, Rahil Doshi, Nitya Khetarpal, Omolola Fatokun, Daphnee Doucet, Ashley Zubkowski, Dorsa Yahya Rayat, Hayley Jackson, Karxena Harford, Afia Anjum, Mahi Zakir, Fei Wang, Siyang

Tian, Brian Lee, Jaanus Liigand, Harrison Peters, Ruo Qi Rachel Wang, Tue Nguyen, Denise So, Matthew Sharp, Rodolfo da Silva, Cyrella Gabriel, Joshua Scantlebury, Marissa Jasinski, David Ackerman, Timothy Jewison, Tanvir Sajed, Vasuk Gautam, and David S. Wishart. DrugBank 6.0: the DrugBank Knowledgebase for 2024. *Nucleic Acids Research*, 52(D1):D1265–D1275, January 2024.

[114] K. Kroenke, R. L. Spitzer, and J. B. Williams. The PHQ-9: validity of a brief depression severity measure. *Journal of General Internal Medicine*, 16(9):606–613, September 2001.

[115] Harlan M. Krumholz. Big Data And New Knowledge In Medicine: The Thinking, Training, And Tools Needed For A Learning Health System. *Health Affairs*, 33(7):1163–1170, July 2014.

[116] J. B. Kruskal. Nonmetric Multidimensional Scaling: A Numerical Method. *Psychometrika*, 29(2):115–129, June 1964.

[117] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, February 2020.

[118] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *The Thirty-Sixth International Conference on Machine Learning (ICML) 2019, Jun 9-15, 2019*, 06 2019.

[119] Han Li, Ruotian Zhang, Yaosen Min, Dacheng Ma, Dan Zhao, and Jianyang Zeng. A knowledge-guided pre-training framework for improving molecular representation learning. *Nature Communications*, 14(1):7568, November 2023. Publisher: Nature Publishing Group.

[120] Li Li, Wei-Yi Cheng, Benjamin S. Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P. Bottinger, and Joel T. Dudley. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine*, 7(311):311ra174, October 2015.

[121] Michelle M. Li, Kexin Huang, and Marinka Zitnik. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering*, 6(12):1353–1369, October 2022.

[122] Andrew Lin, Márton Kolossváry, Jeremy Yuvaraj, Sebastien Cadet, Priscilla A. McElhinney, Cathy Jiang, Nitesh Nerlekar, Stephen J. Nicholls, Piotr J. Slomka, Pál

Maurovich-Horvat, Dennis T. L. Wong, and Damini Dey. Myocardial Infarction Associates With a Distinct Pericoronary Adipose Tissue Radiomic Phenotype: A Prospective Case-Control Study. *JACC: Cardiovascular Imaging*, 13(11):2371–2383, November 2020.

[123] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. 5th International Conference on Learning Representations (ICLR), Poster., May 2017. arXiv:1608.03983 [cs, math].

[124] P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3(1):1236, February 2013.

[125] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[126] Chen Lyu, Bo Chen, Yafeng Ren, and Donghong Ji. Long short-term memory RNN for biomedical named entity recognition. *BMC Bioinformatics*, 18(1):462, October 2017.

[127] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[128] Favour Danladi Makurvet. Biologics vs. small molecules: Drug costs and patient access. *Medicine in Drug Discovery*, 9:100075, 2021.

[129] Miguel Angel Martínez-González, Ana García-Arellano, Estefanía Toledo, Jordi Salas-Salvadó, Pilar Buil-Cosiales, Dolores Corella, Maria Isabel Covas, Helmut Schröder, Fernando Arós, Enrique Gómez-Gracia, Miquel Fiol, Valentina Ruiz-Gutiérrez, José Lapetra, Rosa Maria Lamuela-Raventos, Lluís Serra-Majem, Xavier Pintó, Miguel Angel Muñoz, Julia Wärnberg, Emilio Ros, Ramón Estruch, and PREDIMED Study Investigators. A 14-item Mediterranean diet assessment tool and obesity indexes among high-risk subjects: the PREDIMED trial. *PloS One*, 7(8):e43134, 2012.

[130] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. DeepTox: Toxicity Prediction using Deep Learning. *Frontiers in Environmental Science*, 3, February 2016. Publisher: Frontiers.

[131] Brian McBride. The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 51–65. Springer, Berlin, Heidelberg, 2004.

[132] Clement J McDonald, Stanley M Huff, Jeffrey G Suico, Gilbert Hill, Dennis Leavelle, Raymond Aller, Arden Forrey, Kathy Mercer, Georges DeMoor, John Hook, Warren Williams, James Case, Pat Maloney, and for the Laboratory LOINC Developers. Loinc, a universal standard for identifying laboratory observations: A 5-year update. *Clinical Chemistry*, 49(4):624–633, 04 2003.

[133] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861, September 2018.

[134] Marina Meilă and Hanyu Zhang. Manifold Learning: What, How, and Why. *Annual Review of Statistics and Its Application*, 11(1):393–417, April 2024.

[135] Jörg Menche, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. Uncovering disease-disease relationships through the incomplete interactome. *Science*, 347(6224):1257601, February 2015. Publisher: American Association for the Advancement of Science.

[136] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[137] Giorgio Milazzo, Daniele Mercatelli, Giulia Di Muzio, Luca Triboli, Piergiuseppe De Rosa, Giovanni Perini, and Federico M. Giorgi. Histone Deacetylases (HDACs): Evolution, Specificity, Role in Transcriptional Complexes, and Pharmacological Actionability. *Genes*, 11(5):556, May 2020.

[138] Julie M. Miller, Carlos E. Rochitte, Marc Dewey, Armin Arbab-Zadeh, Hiroyuki Niinuma, Ilan Gottlieb, Narinder Paul, Melvin E. Clouse, Edward P. Shapiro, John Hoe, Albert C. Lardo, David E. Bush, Albert de Roos, Christopher Cox, Jeffery Brinker, and João A.C. Lima. Diagnostic performance of coronary angiography by 64-row ct. *New England Journal of Medicine*, 359(22):2324–2336, 2008.

[139] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002. Publisher: American Association for the Advancement of Science.

[140] Rocco A Montone, Maria Chiara Meucci, and Giampaolo Niccoli. The management of non-culprit coronary lesions in patients with acute coronary syndrome. *European Heart Journal Supplements*, 22(Supplement_L):L170–L175, November 2020.

[141] H. L. Morgan. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5(2):107–113, May 1965. Publisher: American Chemical Society.

[142] Deisy Morselli Gysi, Ítalo do Valle, Marinka Zitnik, Asher Ameli, Xiao Gan, Onur Varol, Susan Dina Ghiassian, J. J. Patten, Robert A. Davey, Joseph Loscalzo, and Albert-László Barabási. Network medicine framework for identifying drug-repurposing opportunities for COVID-19. *Proceedings of the National Academy of Sciences of the United States of America*, 118(19):e2025581118, May 2021.

[143] Holly M. Mortensen, Jonathan Senn, Trevor Levey, Phillip Langley, and Antony J. Williams. The 2021 update of the EPA's adverse outcome pathway database. *Scientific Data*, 8(1):169, July 2021. Publisher: Nature Publishing Group.

[144] Kenneth Morton, Patrick Wang, Chris Bizon, Steven Cox, James Balhoff, Yaphet Kebede, Karamarie Fecho, and Alexander Tropsha. ROBOKOP: an abstraction layer and user interface for knowledge graphs to support question answering. *Bioinformatics (Oxford, England)*, 35(24):5382–5384, December 2019.

[145] Sadako Motoyama, Masayoshi Sarai, Hiroto Harigaya, Hirofumi Anno, Kaori Inoue, Tomonori Hara, Hiroyuki Naruse, Junichi Ishii, Hitoshi Hishida, Nathan D. Wong, Renu Virmani, Takeshi Kondo, Yukio Ozaki, and Jagat Narula. Computed tomographic angiography characteristics of atherosclerotic plaques subsequently resulting in acute coronary syndrome. *Journal of the American College of Cardiology*, 54(1):49–57, 2009.

[146] Diana Navas-Carrillo, Francisco Marín, Mariano Valdés, and Esteban Orenes-Piñero. Deciphering acute coronary syndrome biomarkers: High-resolution proteomics in platelets, thrombi and microparticles. *Critical Reviews in Clinical Laboratory Sciences*, 54(1):49–58, January 2017.

[147] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, February 2004.

[148] P. Ngatchou, A. Zarei, and A. El-Sharkawi. Pareto multi objective optimization. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pages 84–91, 2005.

[149] Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, April 2011. Publisher: Proceedings of the National Academy of Sciences.

[150] Jessica L. Nielson, Shelly R. Cooper, John K. Yue, Marco D. Sorani, Tomoo Inoue, Esther L. Yuh, Pratik Mukherjee, Tanya C. Petrossian, Jesse Paquette, Pek Y. Lum, Gunnar E. Carlsson, Mary J. Vassar, Hester F. Lingsma, Wayne A. Gordon, Alex B. Valadka, David O. Okonkwo, Geoffrey T. Manley, Adam R. Ferguson, and Track-Tbi Investigators. Uncovering precision phenotype-biomarker associations in traumatic brain injury using topological data analysis. *PLOS ONE*, 12(3):e0169490, March 2017. Publisher: Public Library of Science.

[151] Ikujiro Nonaka and Hirotaka Takeuchi. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation.* Oxford University Press, 05 1995.

[152] Stuart Oldham, Ben Fulcher, Linden Parkes, Aurina Arnatkeviciūtė, Chao Suo, and Alex Fornito. Consistency and differences between centrality measures across distinct classes of networks. *PLOS ONE*, 14(7):e0220061, July 2019.

[153] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric Transitivity Preserving Graph Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1105–1114, New York, NY, USA, 2016. Association for Computing Machinery.

[154] Kathleen D. Pagana, Timothy J. Pagana, and Theresa N. Pagana. Mosby's diagnostic and laboratory test reference-E-book. Elsevier Health Sciences, 2014.

[155] Lawrence Page, Sergey Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking : Bringing Order to the Web. In *The Web Conference*, November 1999.

[156] Luigi Palmieri, Rita Rielli, Luca Dematтè, Chiara Donfrancesco, Paola Ciccarelli, Patrizia De Sanctis Caiola, Francesco Dima, Cinzia Lo Noce, Ovidio Brignoli, Alfredo Cuffari, and Simona Giampaoli. CUORE project: implementation of the 10-year risk

score. *European journal of cardiovascular prevention and rehabilitation*, 18(4):642–649, August 2011.

[157] Ian Tong Pan and Joseph D. Romano. Enhancing molecular representation learning through the combination of 3d and 2d graph machine learning (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):29464–29465, Apr. 2025.

[158] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert. Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer's disease. *Medical Image Analysis*, 48:117–130, August 2018.

[159] Vimla L. Patel, Edward H. Shortliffe, Mario Stefanelli, Peter Szolovits, Michael R. Berthold, Riccardo Bellazzi, and Ameen Abu-Hanna. The coming of age of artificial intelligence in medicine. *Artificial Intelligence in Medicine*, 46(1):5–17, May 2009.

[160] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[161] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '14, pages 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[162] Alexander Platzer. Visualization of SNPs with t-SNE. *PloS One*, 8(2):e56883, 2013.

[163] Tiffany M. Powell-Wiley, Yvonne Baumer, Foster Osei Baah, Andrew S. Baez, Nicole Farmer, Christa T. Mahlobo, Mario A. Pita, Kameswari A. Potharaju, Kosuke Tamura, and Gwenyth R. Wallen. Social determinants of cardiovascular disease. *Circulation Research*, 130(5):782–799, 2022.

[164] Xingqin Qi, Eddie Fuller, Qin Wu, Yezhou Wu, and Cun-Quan Zhang. Laplacian centrality: A new centrality measure for weighted networks. *Information Sciences*, 194:240–253, July 2012.

[165] Arwa B. Raies and Vladimir B. Bajic. In silico toxicology: computational methods for the prediction of chemical toxicity. *Wiley Interdisciplinary Reviews. Computational Molecular Science*, 6(2):147–172, March 2016.

[166] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, August 2014. Publisher: Nature Publishing Group.

[167] Ricardo Ramirez, Yu-Chiao Chiu, Allen Hererra, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. Classification of Cancer Types Using Graph Convolutional Neural Networks. *Frontiers in Physics*, 8, June 2020. Publisher: Frontiers.

[168] Jiahua Rao, Shuangjia Zheng, Yutong Lu, and Yuedong Yang. Quantitative evaluation of explainable graph neural networks for molecular property prediction. *Patterns*, 3(12):100628, December 2022.

[169] Jiahua Rao, Shuangjia Zheng, Yutong Lu, and Yuedong Yang. Quantitative evaluation of explainable graph neural networks for molecular property prediction. *Patterns*, 3(12):100628, Dec. 2022.

[170] RDKit: Open-source cheminformatics, (q1 2024) release. https://www.rdkit.org.

[171] Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, and Pascal Friederich. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):1–18, November 2022. Publisher: Nature Publishing Group.

[172] Brad Reisfeld and Arthur N. Mayeno. What is computational toxicology? *Methods in Molecular Biology (Clifton, N.J.)*, 929:3–7, 2012.

[173] Sungmin Rhee, Seokjun Seo, and Sun Kim. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 3527–3534. AAAI Press, 2018.

[174] Ann M. Richard, Ruili Huang, Suramya Waidyanatha, Paul Shinn, Bradley J. Collins, Inthirany Thillainadarajah, Christopher M. Grulke, Antony J. Williams, Ryan R. Lougee, Richard S. Judson, Keith A. Houck, Mahmoud Shobair, Chihae Yang, James F. Rathman, Adam Yasgar, Suzanne C. Fitzpatrick, Anton Simeonov, Russell S. Thomas, Kevin M. Crofton, Richard S. Paules, John R. Bucher, Christopher P.

Austin, Robert J. Kavlock, and Raymond R. Tice. The Tox21 10K Compound Library: Collaborative Chemistry Advancing Toxicology. *Chemical Research in Toxicology*, 34(2):189–216, February 2021. Publisher: American Chemical Society.

[175] Ann M. Richard, Richard S. Judson, Keith A. Houck, Christopher M. Grulke, Patra Volarath, Inthirany Thillainadarajah, Chihae Yang, James Rathman, Matthew T. Martin, John F. Wambaugh, Thomas B. Knudsen, Jayaram Kancherla, Kamel Mansouri, Grace Patlewicz, Antony J. Williams, Stephen B. Little, Kevin M. Crofton, and Russell S. Thomas. ToxCast Chemical Landscape: Paving the Road to 21st Century Toxicology. *Chemical Research in Toxicology*, 29(8):1225–1251, August 2016. Publisher: American Chemical Society.

[176] W. S. Richardson, M. C. Wilson, J. Nishikawa, and R. S. Hayward. The well-built clinical question: a key to evidence-based decisions. *ACP journal club*, 123(3):A12–13, 1995.

[177] Raquel Rodríguez-Pérez and Jürgen Bajorath. Interpretation of Compound Activity Predictions from Complex Machine Learning Models Using Local Approximations and Shapley Values. *Journal of Medicinal Chemistry*, 63(16):8761–8777, August 2020. Publisher: American Chemical Society.

[178] David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, May 2010. Publisher: American Chemical Society.

[179] Joseph D. Romano. Comptoxai: A toolkit for ai research in computational toxicology. https://comptox.ai, 2021.

[180] Joseph D. Romano, Yun Hao, and Jason H. Moore. Improving QSAR Modeling for Predictive Toxicology using Publicly Aggregated Semantic Graph Data and Graph Neural Networks. In *Biocomputing 2022*, pages 187–198. WORLD SCIENTIFIC, September 2021.

[181] Joseph D. Romano, Yun Hao, Jason H. Moore, and Trevor M. Penning. Automating Predictive Toxicology Using ComptoxAI. *Chemical Research in Toxicology*, 35(8):1370–1382, August 2022.

[182] Joseph D. Romano, Van Truong, Rachit Kumar, Mythreye Venkatesan, Britney E. Graham, Yun Hao, Nick Matsumoto, Xi Li, Zhiping Wang, Marylyn D. Ritchie, Li Shen, and Jason H. Moore. The Alzheimer's Knowledge Base: A Knowledge Graph for

Alzheimer Disease Research. *Journal of Medical Internet Research*, 26(1):e46777, April 2024. Company: Journal of Medical Internet Research Distributor: Journal of Medical Internet Research Institution: Journal of Medical Internet Research Label: Journal of Medical Internet Research Publisher: JMIR Publications Inc., Toronto, Canada.

[183] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. Place: US Publisher: American Psychological Association.

[184] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[185] Tong Ruan, Liqi Lei, Yangming Zhou, Jie Zhai, Le Zhang, Ping He, and Ju Gao. Representation learning for clinical time series prediction tasks in electronic health records. *BMC Medical Informatics and Decision Making*, 19(8):259, December 2019.

[186] Papri Saha and Debasish Sarkar. Characterization and Classification of ADHD Subtypes: An Approach Based on the Nodal Distribution of Eigenvector Centrality and Classification Tree Model. *Child Psychiatry and Human Development*, 55(3):622–634, June 2024.

[187] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, 33(11):2881–2907, October 2021.

[188] Eric W Sayers, Evan E Bolton, J Rodney Brister, Kathi Canese, Jessica Chan, Donald C Comeau, Ryan Connor, Kathryn Funk, Chris Kelly, Sunghwan Kim, Tom Madej, Aron Marchler-Bauer, Christopher Lanczycki, Stacy Lathrop, Zhiyong Lu, Francoise Thibaud-Nissen, Terence Murphy, Lon Phan, Yuri Skripchenko, Tony Tse, Jiyao Wang, Rebecca Williams, Barton W Trawick, Kim D Pruitt, and Stephen T Sherry. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 50(D1):D20–D26, January 2022.

[189] Nalini Schaduangrat, Samuel Lampa, Saw Simeon, Matthew Paul Gleeson, Ola Spjuth, and Chanin Nantasenamat. Towards reproducible computational drug discovery. *Journal of Cheminformatics*, 12(1):9, January 2020.

[190] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy,

Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 593–607, Cham, 2018. Springer International Publishing.

[191] Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert Hoog, Nigel Shadbolt, Walter Velde, and Bob Wielinga. *Knowledge Engineering and Management - The CommonKADS Methodology*, volume 24. MIT Press, January 2001.

[192] Lukas Schwingshackl, Jakub Morze, and Georg Hoffmann. Mediterranean diet and health status: Active ingredients and pharmacological mechanisms. *British Journal of Pharmacology*, 177(6):1241–1257, March 2020.

[193] SCORE2-OP working group and ESC Cardiovascular risk collaboration. SCORE2-OP risk prediction algorithms: estimating incident cardiovascular event risk in older persons in four geographical risk regions. *European Heart Journal*, 42(25):2455–2467, July 2021.

[194] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *Medical Image Analysis*, 88:102802, August 2023.

[195] Edward H. Shortliffe. Mycin: A Knowledge-Based Computer Program Applied to Infectious Diseases. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 66–69, October 1977.

[196] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. *Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition*. The Eurographics Association, 2007. Accepted: 2014-01-29T16:52:11Z ISSN: 1811-7813.

[197] Yara Skaf and Reinhard Laubenbacher. Topological data analysis in biomedicine: A review. *Journal of Biomedical Informatics*, 130:104082, June 2022.

[198] SNOMED International. SNOMED CT. https://www.snomed.org, 2024. Accessed: 2025-05-05.

[199] Thereza A. Soares, Ariane Nunes-Alves, Angelica Mazzolari, Fiorella Ruggiu, Guo-Wei Wei, and Kenneth Merz. The (Re)-Evolution of Quantitative Structure–Activity Relationship (QSAR) Studies Propelled by the Surge of Machine Learning Methods. *Journal of Chemical Information and Modeling*, 62(22):5317–5320, November 2022. Publisher: American Chemical Society.

[200] Michelle W. Y. Southey and Michael Brunavs. Introduction to small molecule drug discovery and preclinical development. *Frontiers in Drug Discovery*, 3, November 2023. Publisher: Frontiers.

[201] Jakob Steinfeldt, Thore Buergel, Lukas Loock, Paul Kittner, Greg Ruyoga, Julius Upmeier Zu Belzen, Simon Sasse, Henrik Strangalies, Lara Christmann, Noah Hollmann, Benedict Wolf, Brian Ference, John Deanfield, Ulf Landmesser, and Roland Eils. Neural network-based integration of polygenic and clinical information: development and validation of a prediction model for 10-year risk of major adverse cardiac events in the UK Biobank cohort. *The Lancet. Digital Health*, 4(2):e84–e94, February 2022.

[202] Daniel J. Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011.

[203] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A Deep Learning Approach to Antibiotic Discovery. *Cell*, 180(4):688–702.e13, February 2020.

[204] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, Bette Liu, Paul Matthews, Giok Ong, Jill Pell, Alan Silman, Alan Young, Tim Sprosen, Tim Peakman, and Rory Collins. UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, March 2015.

[205] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Presented at the International Conference on Learning Representations*, September 2018.

[206] Vladimir Svetnik, Andy Liaw, Christopher Tong, J. Christopher Culberson, Robert P. Sheridan, and Bradley P. Feuston. Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6):1947–1958, 2003.

[207] Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L. Gable, Tao Fang, Nadezhda T. Doncheva, Sampo

Pyysalo, Peer Bork, Lars J. Jensen, and Christian von Mering. The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, January 2023.

[208] Kazuhisa Takamura, Shinichiro Fujimoto, Takeshi Kondo, Makoto Hiki, Yuko Kawaguchi, Etsuro Kato, and Hiroyuki Daida. Incremental Prognostic Value of Coronary Computed Tomography Angiography: High-Risk Plaque Characteristics in Asymptomatic Patients. *Journal of Atherosclerosis and Thrombosis*, 24(11):1174–1185, 2017.

[209] Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella PÃ©rez, Matteo Caorsi, Anibal M. Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda: : A topological data analysis toolkit for machine learning and data exploration. *Journal of Machine Learning Research*, 22(39):1–6, 2021.

[210] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000. Publisher: American Association for the Advancement of Science.

[211] Raymond R. Tice, Christopher P. Austin, Robert J. Kavlock, and John R. Bucher. Improving the human hazard characterization of chemicals: a Tox21 update. *Environmental Health Perspectives*, 121(7):756–765, July 2013.

[212] Tox21 protocol of caspase-glo 3/7 cho-k1 cell-based assay for high-throughput screening.

[213] Toxicology in the 21st century data, 2025. https://tripod.nih.gov/pubdata.

[214] Tox21 operational model, 2025. https://tox21.gov/overview/operational-model/.

[215] V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, March 2019. Publisher: Nature Publishing Group.

[216] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2071–2080. PMLR, June 2016. ISSN: 1938-7228.

[217] Jengnan Tzeng, Henry Horng-Shing Lu, and Wen-Hsiung Li. Multidimensional scaling for large genomic data sets. *BMC Bioinformatics*, 9(1):179, April 2008.

[218] Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, and Sam W. Mangham. Kepler mapper: A flexible python implementation of the mapper algorithm. *Journal of Open Source Software*, 4(42):1315, 2019.

[219] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[220] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[221] Sascha Venturelli, Alexander Berger, Alexander Böcker, Christian Busch, Timo Weiland, Seema Noor, Christian Leischner, Sabine Schleicher, Mascha Mayer, Thomas S. Weiss, Stephan C. Bischoff, Ulrich M. Lauer, and Michael Bitzer. Resveratrol as a Pan-HDAC Inhibitor Alters the Acetylation Status of Jistone Proteins in Human-Derived Hepatoblastoma Cells. *PLoS ONE*, 8(8):e73097, August 2013.

[222] F Vitali, S Marini, D Pala, A Demartini, S Montoli, A Zambelli, and R Bellazzi. Patient similarity by joint matrix trifactorization to identify subgroups in acute myeloid leukemia. *JAMIA Open*, 1(1):75–86, May 2018.

[223] Christiaan Vrints, Felicita Andreotti, Konstantinos C Koskinas, Xavier Rossello, Marianna Adamo, James Ainslie, Adrian Paul Banning, Andrzej Budaj, Ronny R Buechel, Giovanni Alfonso Chiariello, Alaide Chieffo, Ruxandra Maria Christodorescu, Christi Deaton, Torsten Doenst, Hywel W Jones, Vijay Kunadian, Julinda Mehilli, Milan Milojevic, Jan J Piek, Francesca Pugliese, Andrea Rubboli, Anne Grete Semb, Roxy Senior, Jurrien M ten Berg, Eric Van Belle, Emeline M Van Craenenbroeck, Rafael Vidal-Perez, Simon Winther, and ESC Scientific Document Group. 2024 ESC Guidelines for the management of chronic coronary syndromes: Developed by the task force for the management of chronic coronary syndromes of the European Society of Cardiology (ESC) Endorsed by the European Association for Cardio-Thoracic Surgery (EACTS). *European Heart Journal*, 45(36):3415–3537, September 2024.

[224] Moritz Walter, Samuel J. Webb, and Valerie J. Gillet. Interpreting Neural Network Models for Toxicity Prediction by Extracting Learned Chemical Features. *Journal of*

Chemical Information and Modeling, 64(9):3670–3688, May 2024. Publisher: American Chemical Society.

[225] Mengyuan Wang, Haiying Wang, and Huiru Zheng. A Mini Review of Node Centrality Metrics in Biological Networks. *International Journal of Network Dynamics and Intelligence*, pages 99–110, December 2022.

[226] Yang Wang, Zengru Di, and Ying Fan. Identifying and Characterizing Nodes Important to Community Structure Using the Spectrum of the Graph. *PLoS ONE*, 6(11):e27418, November 2011.

[227] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, March 2022. Publisher: Nature Publishing Group.

[228] Zifeng Wang, Zichen Wang, Balasubramaniam Srinivasan, Vassilis N. Ioannidis, Huzefa Rangwala, and RISHITA ANUBHAI. Biobridge: Bridging biomedical foundation models via knowledge graphs. In *The Twelfth International Conference on Learning Representations*, 2024.

[229] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, February 1988. Publisher: American Chemical Society.

[230] B Yu Weisfeiler and A A Leman. THE REDUCTION OF A GRAPH TO CANONICAL FORM AND THE ALGEBRA WHICH APPEARS THEREIN. *Nauchno-Technicheskaya Informatsia*, 9:12–16, 1968.

[231] Geemi P. Wellawatte, Heta A. Gandhi, Aditi Seshadri, and Andrew D. White. A Perspective on Explanations of Molecular Prediction Models. *Journal of Chemical Theory and Computation*, 19(8):2149–2160, April 2023.

[232] Michelle Whirl-Carrillo, Rachel Huddart, Li Gong, Katrin Sangkuhl, Caroline F. Thorn, Ryan Whaley, and Teri E. Klein. An Evidence-Based Framework for Evaluating Pharmacogenomics Knowledge for Personalized Medicine. *Clinical Pharmacology & Therapeutics*, 110(3):563–572, 2021. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpt.2350.

[233] Daniel S. Wigh, Jonathan M. Goodman, and Alexei A. Lapkin. A review of molecular representation in the age of machine learning.

WIREs Computational Molecular Science, 12(5):e1603, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1603.

[234] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, January 2018. Publisher: The Royal Society of Chemistry.

[235] Zhenxing Wu, Jike Wang, Hongyan Du, Dejun Jiang, Yu Kang, Dan Li, Peichen Pan, Yafeng Deng, Dongsheng Cao, Chang-Yu Hsieh, and Tingjun Hou. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature Communications*, 14(1):2585, May 2023. Publisher: Nature Publishing Group.

[236] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, August 2020. Publisher: American Chemical Society.

[237] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[238] Juri Yanase and Evangelos Triantaphyllou. A systematic survey of computer-aided diagnosis in medicine: Past and present developments. *Expert Systems with Applications*, 138:112821, December 2019.

[239] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and L. Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Presented at the International Conference on Learning Representations*, December 2014.

[240] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: generating explanations for graph neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 9244–9255. Curran Associates Inc., Red Hook, NY, USA, 2019.

[241] Haiyuan Yu, Philip M. Kim, Emmett Sprecher, Valery Trifonov, and Mark Gerstein. The Importance of Bottlenecks in Protein Networks: Correlation with Gene Essentiality and Expression Dynamics. *PLOS Computational Biology*, 3(4):e59, April 2007. Publisher: Public Library of Science.

[242] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–19, 2022.

[243] Xiang Yue, Zhen Wang, Jingong Huang, Srinivasan Parthasarathy, Soheil Moosavinasab, Yungui Huang, Simon M Lin, Wen Zhang, Ping Zhang, and Huan Sun. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, 36(4):1241–1251, February 2020.

[244] Jin Zhang, Daniel Mucs, Ulf Norinder, and Fredrik Svensson. LightGBM: An Effective and Scalable Algorithm for Prediction of Chemical Toxicity–Application to the Tox21 and Mutagenicity Data Sets. *Journal of Chemical Information and Modeling*, 59(10):4150–4158, October 2019. Publisher: American Chemical Society.

[245] Xingmin Aaron Zhang, Amy Yates, Nicole Vasilevsky, J. P. Gourdine, Tiffany J. Callahan, Leigh C. Carmody, Daniel Danis, Marcin P. Joachimiak, Vida Ravanmehr, Emily R. Pfaff, James Champion, Kimberly Robasky, Hao Xu, Karamarie Fecho, Nephi A. Walton, Richard L. Zhu, Justin Ramsdill, Christopher J. Mungall, Sebastian Köhler, Melissa A. Haendel, Clement J. McDonald, Daniel J. Vreeman, David B. Peden, Tellen D. Bennett, James A. Feinstein, Blake Martin, Adrianne L. Stefanski, Lawrence E. Hunter, Christopher G. Chute, and Peter N. Robinson. Semantic integration of clinical laboratory tests from electronic health records for deep phenotyping and biomarker discovery. *NPJ digital medicine*, 2:32, 2019.

[246] Juan Zhao, QiPing Feng, Patrick Wu, Roxana A. Lupu, Russell A. Wilke, Quinn S. Wells, Joshua C. Denny, and Wei-Qi Wei. Learning from Longitudinal Data in Electronic Health Record and Genetic Data to Improve Cardiovascular Event Prediction. *Scientific Reports*, 9(1):717, January 2019. Publisher: Nature Publishing Group.

[247] Weifan Zheng and Alexander Tropsha. Novel Variable Selection Quantitative StructureProperty Relationship Approach Based on the k-Nearest-Neighbor Principle. *Journal of Chemical Information and Computer Sciences*, 40(1):185–194, January 2000. Publisher: American Chemical Society.

[248] Zhiqiang Zhong, Cheng-Te Li, and Jun Pang. Hierarchical message-passing graph neural networks. *Data Mining and Knowledge Discovery*, 37(1):381–408, January 2023.

[249] Yini Zhou, Chao Ning, Yijun Tan, Yaqi Li, Jiaxu Wang, Yuanyuan Shu, Songping Liang, Zhonghua Liu, and Ying Wang. ToxMPNN: A deep learning model for small

molecule toxicity prediction. *Journal of Applied Toxicology*, n/a(n/a), 2024. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jat.4591.

[250] Siyi Zhu, Jiaxin Bing, Xiaoping Min, Chen Lin, and Xiangxiang Zeng. Prediction of Drug–Gene Interaction by Using Metapath2vec. *Frontiers in Genetics*, 9, July 2018. Publisher: Frontiers.

[251] Xiaojin Zhu and Zoubin Ghahramani. *Learning from Labeled and Unlabeled Data with Label Propagation*. CiteSeer, 2003.

[252] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.