Universiteit
Leiden

Master Computer Science

Foundational Ontologies to Improve the Explainability of Explainable AI Algorithms

Name:	Rosa Zwart
Student ID:	s2034190
Date:	[12/12/2023]
Specialisation:	Bioinformatics
1st supervisor:	Katy Wolstencroft
2nd supervisor:	Annika Jacobsen

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Abstract

Unlike traditional AI techniques, explainable AI (XAI) removes the ‘black-box’ nature of machine learning algorithms and adds explanations to its decisions. This enables users to subject the predictions to their own reasoning which is essential for decision-makers to assess the reliability of a trained XAI. In this project, we investigated whether the explainability of a drug repurposing XAI algorithm applied on the rare disease Duchenne muscular dystrophy (DMD) can be improved. To attempt this, a conceptual modelling approach has been developed that relies on the use of foundational ontologies to facilitate the conveyance of the conceptual model towards domain experts for reaching consensus. Following this approach, the concepts and relations found in the knowledge graph serving as the input of the XAI algorithm have been aligned to the relevant domain in order to attempt generating explanations that show more recognizable information paths to reason about why a certain prediction has been made. By decomposing the term explainability into multiple aspects and including them in questionnaires, explanations have been compared assessing whether the explainability has improved after aligning the input knowledge graph to the newly designed conceptual model. No significant changes in the explainability of the generated explanations have been found by adding the developed conceptual modelling approach to the drug repurposing pipeline. The findings of this project did lead to a new perspective as they have shown that the explainability might not get improved solely on changing the conceptual structure of the explanations. Instead, the explainability needs to be improved by also taking into account the representation of the explanations which highlights the need for further user-centric development of XAI.

Acknowledgements

First and foremost, I would like to thank my daily supervisor César whose support, guidance and knowledge have been invaluable throughout this challenging but great experience. I would also like to thank Katy and Annika for their amazing insight and helpful suggestions. I am grateful to have been a part of the LUMC BioSemantics group full of people who were very supportive and inspiring. Finally, a shout out to Karolis, who helped me tremendously in finding many participants for my survey, which was otherwise a very difficult task.

Chapter 1

Introduction

1.1 Explainable AI

In this day and age, the collection and access of vast amounts of data are indispensable across many fields in science. The sheer quantity of data necessitates researchers to utilize tools that are capable working with these enormous amounts of information and subsequently yield insight from it. In the last decade, the use of machine learning as such a tool has massively increased in popularity due to its proven ability [16] to learn from large quantities of data catching complex correlations that are beyond human comprehension. Although the predictive power of AI methods is advancing and machine learning applications are increasingly prevalent across many domains, the ability for humans to understand how an AI model arrives at a certain prediction, lags behind. For example, a deep learning model consisting of many layers has to adjust its millions of parameters in order to deliver a prediction about the given dataset. The high-complexity causes deep learning models to have a "black-box" nature. It poses an issue when "black-box" predictors are being used for solving problems in high-risk and high-impact fields such as healthcare [5].

To fully realize the potential of AI techniques on high-impact failure-sensitive tasks, a new field has arisen that conducts research on explainable AI (XAI). Developed XAI methods aim to remove the "black-box" nature of machine learning by adding explanations to its decisions. This enables users to subject the predictions to their own reasoning and knowledge which is essential for decision-makers to assess the reliability of the predictions from a XAI.

1.2 Drug Repurposing for Duchenne Muscular Dystrophy

In a previous project, a method is proposed which "obtains drug candidates that can be used to treat symptoms related to the rare disease Duchenne muscular dystrophy (DMD)" [57]. A XAI method has been applied in order to generate explanations that can be assessed by clinicians and researchers. In this way, experts can accept or reject the yielded predictions by evaluating the explanations using their knowledge and reasoning.

The disease DMD is classified as a rare disease affecting 1 in 3,600 male live-born infants. Due to the disease being X-linked recessive, male infants are more frequently afflicted in comparison to female infants. This rare disease is caused by a mutation in the dystrophin gene *DMD* which is one of the largest genes found in the human genome. The disease does not have to be inherited and can be caused by a new mutation in this gene instead. *DMD* provides the production of the protein dystrophin which is severely impeded or even disabled by duplications, deletions or point mutations in the gene sequence. Dystrophin deficiency causes degeneration of muscle fiber leading to cardiac and orthopedic complications. Patients have a low life expectancy, less than thirty years, as the disease causes fatal problems resulting from progressive muscle loss such as respiratory muscle weakness and cardiomyopathy. Currently, there is no known treatment that can stop the progression of this disease. [64]

Development of new treatment for people that suffer from this disease is difficult and unappealing as pharmaceutical companies need to recoup the costs of research and clinical trials for a new orphan drug[1] from the relatively small pool of patients that suffer from DMD [37]. While research is done on new treatments for DMD, clinicians also attempt to treat the symptoms that are found in the clinical picture of this disease.

Datasets are accessible that provide researchers with information about drugs already marketed and used for treating certain symptoms such as the online drug information resource DrugCentral [62]. Also, biomedical and genomic data can be retrieved from many different data sources that are easily accessed using for example the National Center for Biotechnology Information (NCBI) services [2]. These large amounts of data generated from already completed research can be utilized to explore possible new links between a marketed drug and a symptom that is related to the rare disease DMD applying XAI techniques.

[1] An orphan drug is defined as a drug that is used to treat or prevent a life-threatening rare disease or a drug that is expected to generate insufficient profit which does not justify the costs of the development and research of the treatment.

1.3 Project Objective

For explanations generated by XAI to be interpretable and trusted by users, it is important that they are conceptually clear and sound. Users might perceive an explanation to have better explainability when reasoning patterns are included that are favoured and often used by the users with expertise in the relevant domain of the problem. Thus, a conceptual model needs to be built that represents the domain knowledge as accurately as possible while it is adjusted to the available concepts and relations in the input data. By using a Foundational Ontology, we facilitate the process of arriving at a conceptual model that might increase explainability when the input data is aligned to it. This is because a Foundational Ontology provides well-defined domain-independent conceptual primitives that enable precise and consistent expression of the model.

Via ontology-based conceptual modelling, the improvement in the structure of the data would not only improve explainability, but also support the FAIR principles [27]. These FAIR guiding principles have been set up to maximize the value of shared data, covering the notions *Findability*, *Accessibility*, *Interoperability* and *Reusability* [66]. Providing a clear and precise conceptual structure of the data allows for interoperability. Due to aligning the dataset to the created conceptual model, new users of the data will have a facilitated understanding about how new concepts and relations can be harmonized with the existing conceptual structure.

Given these findings, the following research question is asked that we attempt to answer during this project: *Does the use of Foundational Ontologies improve the explainability of explainable AI algorithms?*

The research question will be answered by revisiting the drug repurposing pipeline on symptoms of the rare disease DMD developed in the project [57] mentioned in the previous section (Section 1.2). This project is referred to as the eXplainable AI DMD Drug Repurposing (XAI-DMD-DR) project throughout the remainder of the thesis. In this current project, we investigate whether the explainability of the explanations generated by the XAI algorithm can be improved by enhancing the structure of the input data. Deciding which changes need to be made is enabled due to applying a Foundational Ontology which has not been done in the XAI-DMD-DR project. The explanations are compared that result from the XAI drug repurposing pipeline with the original dataset from the previous project as input and with the restructured dataset as input. Based on this comparison focusing on the explainability of the explanations, we come to a conclusion for the research question.

Chapter 2

Background Information

2.1 Ontologies

The term “ontology” is often associated with metaphysics, which is one of the disciplines in philosophy studying the fundamental nature of reality. Nowadays, this term appears increasingly often in computer science literature due to the growing interest in giving well-defined meanings to the vast amount of data that has become available on the World Wide Web. An “ontology” can be defined as “a formal, explicit specification of a shared conceptualization”[23, 61]. In context of ontologies used in the field of computer science, we often talk about conceptual models [18]. A conceptual model has the purpose of clearing out ambiguities and improving the knowledge representation within a dataset.

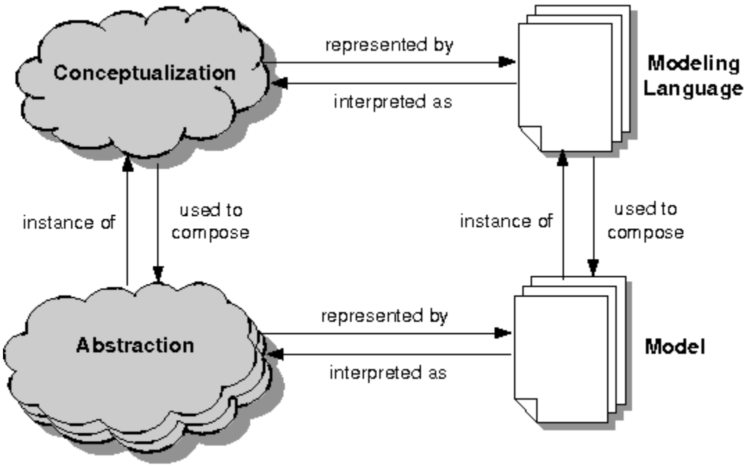


Figure 2.1: The diagram that shows the relations between “conceptualization”, “abstraction”, “modelling language” and “model” taken from [26].

In Figure 2.1 the relations are shown that exist between “conceptualization”, “abstraction”, “modelling language” and “model”. Conceptualization and abstraction are phenomena existing in the minds of users. With conceptualization it is meant that a user forms ideas or principles in their mind which is used to represent their idea about any domain by cognitive processes such as categorization and reasoning. This idea about how a specific domain is structured and what all entities in this domain mean, forms an abstraction. The abstraction created by one user compared to another about the same domain can differ as humans often have varying views on reality that conflict with each other in various degrees. Here, the need of a conceptual model arises that can communicate one's abstraction about a certain domain to others. This model is composed by a modelling language that allows representing all elements in a concise, complete and unambiguous way [26].

2.2 Foundational Ontologies

There exist different types of ontologies as they differ in types of domains that are represented by them as well as the level of formalism provided by the ontology. The different ontologies are classified based on the specificity level or for which purpose they are used [18]. This classification is shown in Figure 2.2 starting with the top level ontologies. Top level ontologies are also known as foundational ontologies [18]. These ontologies describe the most general concepts such as objects, events and provide fundamental types of relations. Thus, the concepts and relation types given by a foundational ontology are domain-independent. Domain ontologies and task ontologies utilize the terms that are given by top level ontologies in order to describe a generic domain or task, respectively. Application ontologies specialize both domain and task ontologies to describe concepts that depend on a specific domain and a task [24]. To exemplify, a domain ontology might be developed to capture the concepts and relations that describe the field of medicine, including concepts such as diseases, symptoms and treatments. A task ontology related to the medical field might represent concepts such as diagnostics and monitoring patients. In the same context, an application ontology could describe the requirements for a specific healthcare system which will include concepts and relations from domain and task ontologies such as the mentioned examples.

Foundational ontologies provide generic modelling aspects that can be applied during ontology development [39]. There exist different foundational ontologies that typically contain some way of categorizing all entities as well as specifying the nature of the relationships that are found between them. Some examples of foundational ontologies are Unified Foundational Ontology (UFO) [28], Basic Formal Ontology (BFO) [60], General Formal Ontology (GFO) [47] and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [11].

By having these generic building blocks with which an domain-specific ontology can be built, the interoperability between ontologies can be greatly improved [10]. This is because ontologies built using the same foundational on-

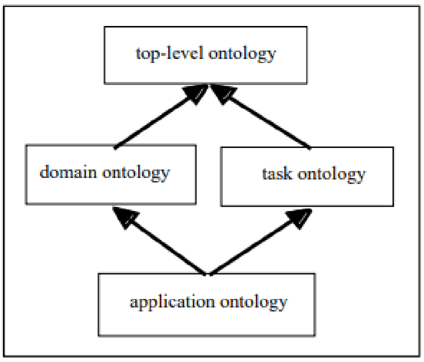


Figure 2.2: The diagram showing the classification of ontologies based on specificity level where the arrows show the specialization relationships taken from [24].

tology share the same domain-independent conceptual components and rules, improving on interpretability of the models despite being domain agnostic artefacts. Parts from one ontology can be incorporated into other related ontologies as modules, introducing facilitated reusability. Due to the well-defined modelling elements of foundational ontologies, they also provide a precise and unambiguous way of describing the types of concepts and relations present in the domain. In this way, the quality of the designed ontology can be guaranteed.

2.3 Unified Foundational Ontology

As a top level ontology, UFO provides a set of basic categories and relations that all have a clear and consistent meaning. These clear definitions and distinctions between the ontological elements allow for building specialized ontologies in a robust and unambiguous way.

UFO is known for its fundamental distinction between endurants and perdurants, where endurants are individuals that exist in time without loosing their identity when it goes through qualitative changes (e.g., objects). Perdurants are individuals that are an accumulation of temporal parts that have unfolded throughout time (e.g., processes). This distinction allows for classification of entities based on their temporal persistence [28]. Endurants can vary in rigidity for which we have the classes rigid, anti-rigid and semi-rigid. Rigid endurants stay unchanged over time while anti-rigid endurants will cease to exist. For example, a person cannot stop being a person without ceasing to exist. However, a person can be a student for a certain amount of time without this person losing their identity. Semi-rigid endurants are entities that stay unchanged over time for some and can cease to exist for others. Take as an example a music band that will cease to exist when it does not have any music artist as member while a person can stop being a music artist and still exists.

Another distinction that UFO follows is between substantial and dependent entities. The former can exist independently while the latter depend on the existence of other entities.

For the different relationship classes, UFO enables modelling the multiplicity of relationships between entities as well as the dependency expressing how entities are related to each other and how the relation is constrained.

These are just a few principles provided by UFO. To get an idea about how extensive the theory is behind this foundational ontology, a general overview of the categories for entities described and provided in UFO are shown in the taxonomy tree that includes the categories around endurants and perdurants in Figure 2.3.

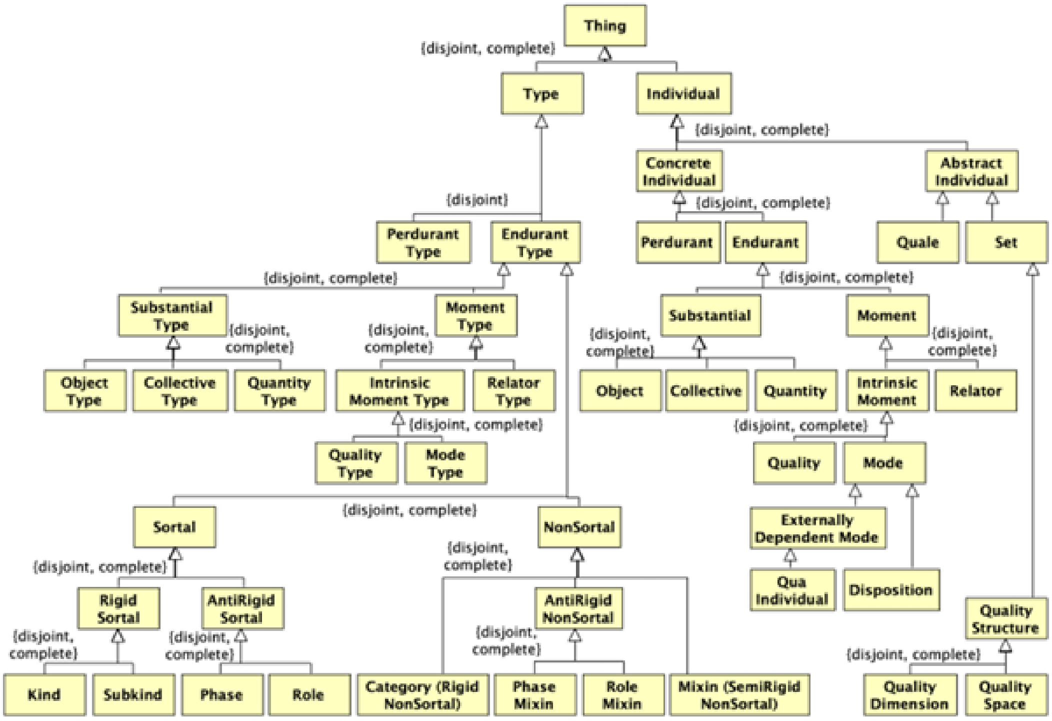


Figure 2.3: The taxonomy of UFO taken from [28].

2.4 OntoUML: An Ontology-based Conceptual Modelling Language

OntoUML is a conceptual modelling language that helps to define concept structures that comply with the theories embedded by UFO, including its categories and axioms [29]. The modelling language facilitates building a domain-specific model since it enables expressing a model in a concise way using well-defined domain-independent concepts and rules.

OntoUML distinguishes between ‘Types’ and ‘Individuals’ which enables classification of entities. ‘Individuals’ are instantiated from ‘Types’ and share characteristics with other instances from the same type. In a model built with OntoUML, only the ‘Types’ are included in the ontology schema while instantiated individuals are not specified. A ‘Type’ can have a generalization relation with other types such that it is a supertype of multiple subtypes or a subtype of a supertype. This means that individuals can instantiate for multiple types. For example, an ‘Individual’ would be Sesam and the ‘Type’ of this entity is a dog.

Another important aspect in OntoUML is whether the ‘Type’ provides the identity principle to an entity or not. The identity principle says that each ‘Individual’ has its own identity carried throughout its whole existence. In our simple example, Sesam would not exist when he stops being a dog. This means that the ‘Type’ dog provides his identity. This identity can be defined using identity conditions such that it is clear when an entity is the same as another entity based on their attribute values. This identity principle adds constraints on how the hierarchy between ‘Types’ can be constructed.

Each ‘Type’ can have a certain stereotype that specifies which additional rules apply to this ‘Type’. For example, when a ‘Type’ provides the identity principle to an entity it can be a ‘Kind’, ‘Collective’ or ‘Relator’. A ‘Kind’ is a construct that represents a functional complex, meaning that the entity is a whole with parts that all contribute differently to its functionality. Concepts that represent entities with an homogeneous internal structure in which each part is perceived the same way are ‘Collectives’. The stereotype ‘Relator’ is used to represent an entity that exists merely to connect two or more individuals to each other via a material relation. Thus, referring back to our example, a dog is made out of many different body cells working together in order to live its life which leads to us considering this ‘Type’ having ‘Kind’ as stereotype. Sesam is a dog owned by a person where this ownership only exists for connecting the dog to the owner. This implies that ownership can be stereotyped as a ‘Relator’.

Another group of stereotypes can contain instances that possibly follow different identity principles. Stereotypes that allow this include ‘Categories’. In our simple example, we could say that dogs as well as cats are domesticated animals. A cat would be a different ‘Kind’ instantiating ‘Individuals’ with different characteristics and identity principles compared to dogs. However, we can categorize cats and dogs using the concept domesticated animals. This means that the ‘Individuals’ of the ‘Kind’ dog and cat will also share some features such as the definition that dogs and cats are both animals suitable for being held as a pet.

Stereotypes, such as ‘Role’ and ‘SubKind’ do not provide the identity principle, but carry them over although the rule applies here that these stereotypes can only be associated to one identity provider. For example, Sesam is a Shiba Inu which is a ‘Subkind’ of the ‘Kind’ dog. This ‘Subkind’ cannot be related to another identity provider. In this case, a Shiba Inu can only be a dog and never be a cat. Here, we also need to address another property of ‘Types’ being rigidity. A ‘Type’ is rigid when an ‘Individual’ instantiates for this ‘Type’ throughout its whole existence. Rigid types thus define characteristics that are essential for their instances. Some rigid stereotypes are ‘Kind’, ‘Subkind’, ‘Relator’, ‘Category’ and ‘Collective’. An ‘Individual’ can stop instantiating for an anti-rigid type while still existing as the characteristics provided by this ‘Type’ are non-essential. An anti-rigid type would for example be ‘Role’. Considering the example again, it can be said that the dog Sesam can stop being a pet without him losing its existence which concludes that pet is a ‘Role’.

In OntoUML, ‘Individuals’ can be tied together by some relation [25]. Each relation has a number of relata which are the entities that are connected to each other. Relations are classified into two categories being the ‘Material’ and ‘Formal’ relations. In ‘Material’ relations, there exists an instance being a ‘Relator’ that mediates the relation between relata. ‘Formal’ relations only need the relata themselves to hold. For a ‘Material’ relation, it can be exemplified by the relation of a pet owner owning a pet. For this relation to hold, there must exist ownership.

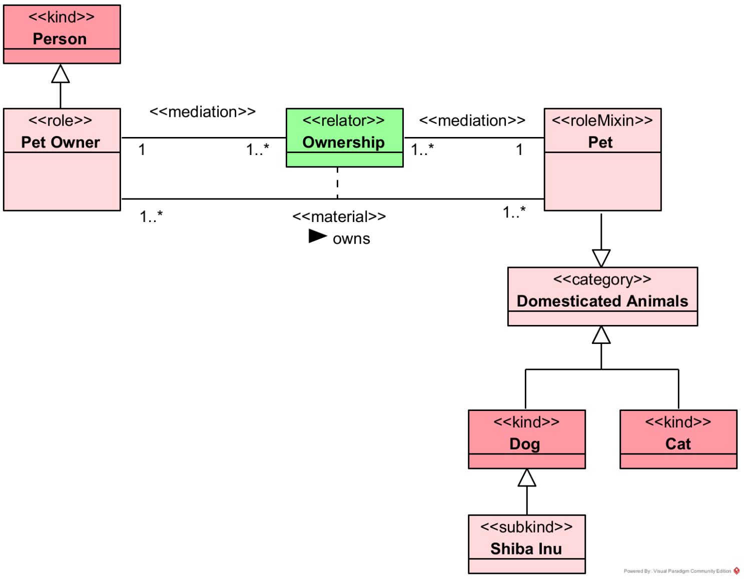


Figure 2.4: The OntoUML schema expressing a simple example scenario about pets to show a few key principles of applying OntoUML to an abstraction.

The OntoUML model resulting from the discussed example is shown in Figure 2.4. There are numerous other distinctions, rules, class and relationship stereotypes provided by OntoUML. However, in this thesis only the elements have been discussed and exemplified that are the most prominent in the conceptual models designed throughout this project and to give an idea about the general principles of the application of OntoUML.

2.5 Goal Modelling

Goal modelling is a method often used in software engineering, a field that necessitates a lot of attention to the elicitation, specification, analysis and validation of software requirements [19]. This method allows to create a hierarchical model that shows what goals need to be reached according to all stakeholders and organizes how these goals are achieved.

2.5.1 iStar Modelling Language

For goal modelling in this project, the *i** modelling language is used which adapts a lot of similarities to conceptual modelling languages as it focuses on intentional, social and strategic dimensions [15]. It can answer questions such as *why* goals need to be achieved, *who* requires which objectives and *how* these goals are achieved. Thus, the *i** modelling language is goal-oriented as well as actor-oriented. To give an idea about the usage of this goal modelling framework, a summary will be given of the guideline of the iStar tool [15] discussing the elements that are used during this project.

The first class of entities included in this framework is *actors*. *Actors* are active and autonomous who aim at achieving goals. This class can be subdivided into *role* and *agent* where *role* is used for an abstract characterization and *agent* represents an entity that has a physical or concrete manifestation. Distinguishing between these two classes of *actors* is not obligatory when these types are not relevant for the scenario that is modelled. In the framework, actors are graphically depicted as circles shown in Figure 2.5. All the intentions of an actor are contained inside the *actor boundary*. In this grey area, all elements and relationships related to the intentions of that actor are placed (Fig. 2.6).

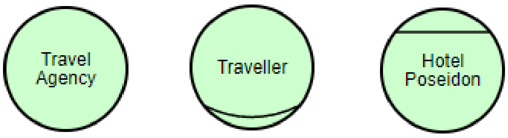


Figure 2.5: Examples of an actor, role and agent using the graphical notation of iStar framework.

Actors can have relations between each other. Using the *is-a* actor link, *roles* can be specializations into other *roles*. For example, a traveller is either a traveller by train or plane. The second actor link is *participates-in* that relates two actors representing any kind of association except for specialization. Due

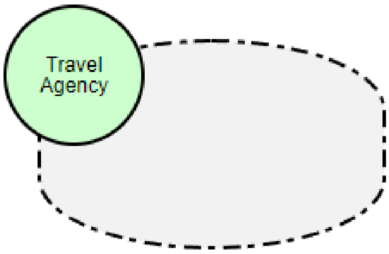


Figure 2.6: Example of an actor boundary using the graphical notation of iStar framework.

to this broad representation, multiple meanings can be attached to this type of link and depends on the context of the scenario. It could be said that it connects between an *agent* and *role* indicating a *plays* relationship. To exemplify, the hotel named Hotel Poseidon in the current scenario can play a role as an all-inclusive resort. It is also possible to have a *part-of* relation where both relata are of the same *actor* type. We can connect the *agent* swimming pool to Hotel Poseidon to show that this swimming pool is part of the hotel. These relations and given examples are shown in Figure 2.7.

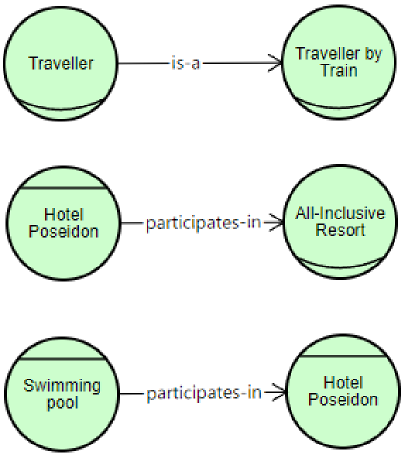


Figure 2.7: Examples of actor association links using the graphical notation of iStar framework.

Actors have intentions which are things that they want. Requirements that are demanded by a certain *actor* are represented with intentional elements contained by its *actor boundary*. One of these elements is the *goal* which is a state that needs to be achieved and can only be reached when its defined criteria have been met. A *goal* would be to book tickets for a train which has a clear criterium that the traveller receives their ticket. The element *quality* is a state that is desired to be achieved at some level. For example, the comfort of hotel guests, which can be achieved at different levels. The intentional element *task* is an action that needs to be performed in order to achieve a goal such as paying for a ticket. Lastly, *resource* refers to the entity that is required to perform a certain task. A ticket can only be paid using some kind of payment method.

Intentional elements can be connected to each other using one of four types of links being *refinement*, *needed-by*, *contribution* and *qualification*. The first link, *refinement*, relates one intentional element as the parent to one or more children elements. A *refinement* can be an *AND-refinement* or *OR-refinement*. The former says that the parent intentional element is fulfilled when all its children elements are fulfilled. For a parent linked to children via an *OR-refinement*, the parent is fulfilled in the case that at least one of the child elements is fulfilled. Using our examples, tickets can only be received by a traveller when they are booked and then sent by the relevant travel agency. This is indicated by the *AND-refinement*. For the *OR-refinement* link, we could say that as soon as the traveller has paid for the tickets, the goal of booking a ticket has been achieved.

The *needed-by* relationship means that a *resource* is needed to perform a *task* to which it is linked. Paying for tickets can be performed when there is a payment method at hand.

Qualities can be linked to intentional elements with the *contribution* links. A *quality* is considered fulfilled or satisfied when there is sufficient positive evidence. In the case of existing negative evidence, a *quality* can be denied. There are multiple types for *contribution* links for the different levels of positive or negative evidence it provides. The types *make* and *help* indicate that the intentional element gives positive evidence for the fulfilment of the parent *quality* to a sufficient or partial degree, respectively. This is the same case for the types *break* and *hurt*. However, they bring evidence against the fulfilment of the target leading to denial of the *quality* when there is sufficient negative evidence. For example, providing comfortable beds and quality food helps to have comfortable hotel guests while a crowded swimming pool counteracts.

Linking a *quality* to another intentional element with the *qualification* relation, it is expressed that the achievement of the target intentional element needs to be performed with a desired quality in mind. For instance, if the all-inclusive resort is to achieve its goal of increasing profits, it must do so while providing the guests a high level of comfort.

In Figure 2.8, the goal models are shown that result from the given examples.

2.6 Knowledge Graphs

Information is often stored in tabular data. In each row of a data table, an identifier is stored along with the attributes of this data entry. A table is re-

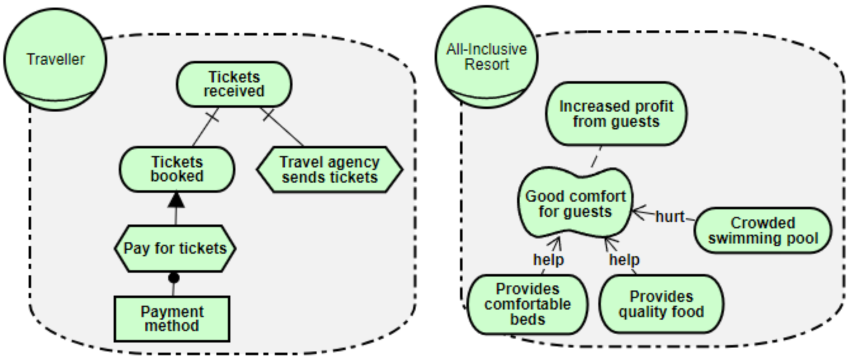


Figure 2.8: Examples of intentional elements and their links using the graphical notation of iStar framework.

stricted to storing entries with the same properties. To allow relations between different data points, a relational database is used. In these databases, it is difficult to add new categories as you need to add new tables with properties that comply with the added categories. Relations between these tables need to be predefined before queries can be performed. In contrast to tabular data, knowledge graphs allow for the addition of new data and relationships without needing any changes to the structural schema of the database. That is why knowledge graphs offer greater flexibility for structuring data than tabular data. Therefore, when handling conceptually rich and variable data, the preferred option would be to employ a knowledge graph as data structure.

A graph is a data structure consisting of nodes and edges. A node represents an entity while an edge links one node with another node expressing the existence of some relation between the entities. A graph can be described as being directed or undirected. For directed graphs, all edges have a specific direction leading from one node to another. When there is no direction specified for each edge, the graph is categorized as undirected. The relations between entities in an undirected graph are reciprocal while there is a clear object and subject role for relations expressed in directed graphs.

A knowledge graph [33] is a data structure that is suitable for representing data that is rich with different concepts and relations in a concise and intuitive way. This is enabled due to all nodes and edges being labelled. A graph with nodes and edges of different types is generally known as a heterogeneous graph, as opposed to homogeneous graphs which consist of a single type of node and a single type of edge. Using the various types for expressing classes of entities and different relations, the conceptual model to which the data is aligned can be retained while the data instances are being stored in a graph data structure. The differences between an undirected homogeneous graph and a directed knowledge graph are illustrated in Figure 2.9.

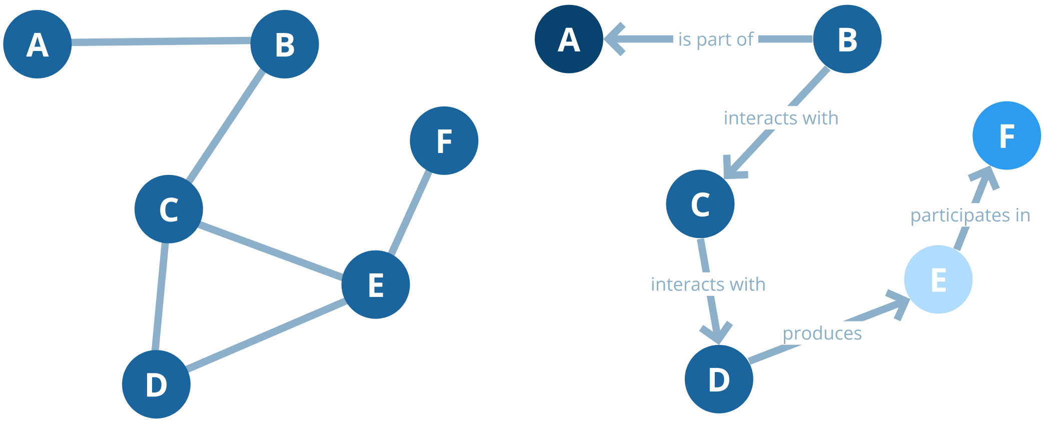


Figure 2.9: At the left, an example of an undirected graph is shown in which edges and nodes have one type. At the right, a directed knowledge graph is illustrated with four node types and four edge types present. The node types are distinguished by colour. The different edge types can be discerned based on their edge labels.

2.7 Graph-based Deep Learning

Particularly in the bioinformatics and biomedical fields, interest in the research on graph neural networks (GNNs) has grown [55]. One of the reasons would be that the application of GNNs enables integration of knowledge graphs into the prediction pipeline. This allows for conceptual enrichment added to the training of a machine learning algorithm.

For graph learning tasks there are three types of tasks that are often performed [68]. The first tasks are at node-level, including node classification. Another group contains edge-level tasks referring to edge classification or link prediction. Lastly, there are graph-level tasks such as graph classification and graph matching.

For the drug repurposing task during this project, link prediction will be performed by the implemented GNN model in order to find potential connections between drugs and symptoms that are related to the disease DMD.

For neural networks (NNs) that accept grid-structured data such as tabular data, sequences and images, all input samples go through the exact same architecture. In contrast to this, the architecture of a GNN changes for each node. In a GNN model, node representations are iteratively updated which is done by passing and aggregating messages of neighbouring nodes [42]. First, there is the message passing part of the model, during which a message vector for each neighbour node is acquired. There exists multiple ways to calculate these message vectors ranging from simple operations to neural networks. Now, the message vectors of each neighbouring node are aggregated resulting into a single vector that will have captured for example topological information from the local neighbourhood of a node. There are a lot of options for the aggregation function such as taking the mean, using the summation or max-pooling over the message vectors. For the update function, the current representation of each node is combined with the aggregated messages extracted from its neighbours which results in a new node representation. An illustration of the architecture of a GNN model given a simple example of a graph can be found in Figure 2.10.

In a supervised learning setting, after each batch of input, the GNN trains using gradient-based optimization methods that will adjust the model parameters such that the difference between the predicted and true values is minimized.

Adapting the architecture of the GNN model will adjust the size of the neighbourhood on which the message-passing process is performed. As each layer of the architecture represents a new message-passing process, an added layer will find the neighbouring nodes of the previously found neighbours. Thus, using a total of two layers in the GNN model architecture enables capturing information from the nodes that are two hops away.

2.8 Explainable AI Methods

In order to justify the use of AI methods for performing tasks from domains in which ethical regulations play a major role, their outcomes need to be verified by human decision makers. This would be relevant for fields such as healthcare where decisions need to be made that have direct impact on the well-being of people [49]. By removing the 'black-box' nature of AI, transparency of AI systems can be improved, approaching the development of *trustworthy AI*. *Trustworthy AI* systems need to adhere to lawfulness, ethicality and robustness. This term is defined by the High-Level Group on AI. The European Commission has formed this independent group of experts [6].

To remove this 'black-box' from AI systems, explainable AI methods are being developed and improved. These methods need to give users insight about how an AI model arrives at a certain prediction while the accuracy and robustness of its predictive performance are kept at a high level.

The vast collection of explainable AI methods can be split into two groups being the model-specific and model-agnostic methods [49, 46]. Explanations are generated natively by the AI model itself when referring to model-specific methods. This means that machine learning models are created from scratch, implementing the generation of explanations in the architecture of the models. In this way, the explainability of the AI system aligns very closely to the model architecture which makes the explanations more accurately describe the model. The drawback of a self-explaining model lies in its inferior predictive performance when compared to machine learning models that have undergone extensive performance-driven research [8]. This is why development has been

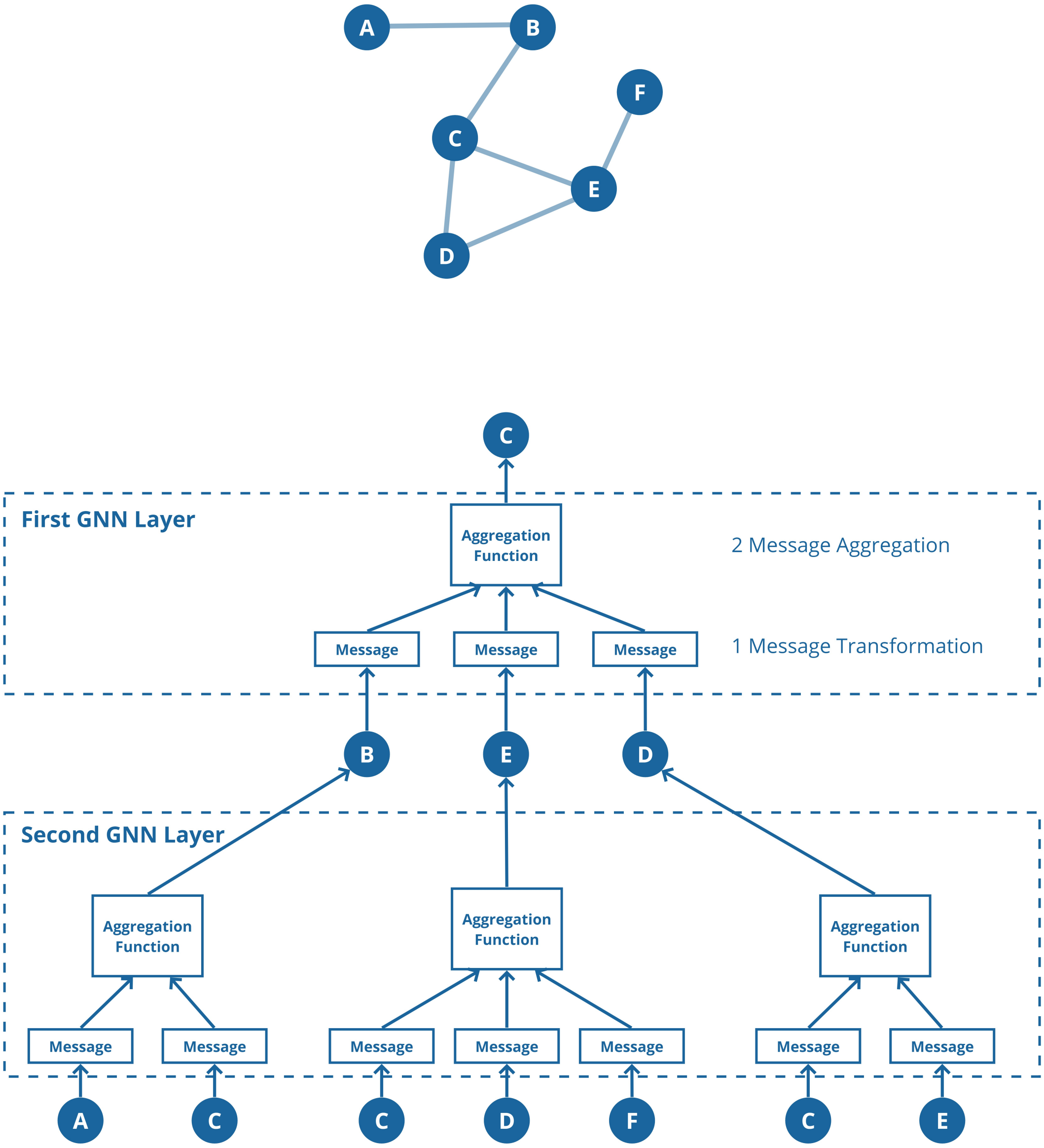


Figure 2.10: A diagram of the architecture of a GNN with two layers illustrating node C as input inspired from diagrams shown in [42]. The nodes in the GNN architecture diagram correspond to the graph above this illustration. Message Transformation: The message passing part ranging from being a simple operation to a neural network. Message Aggregation: The message aggregation part in which the aggregation is often done using a summation, taking the average or max-pooling.

focused on the other category of explainable AI methods being the model-agnostic approaches that can be applied to AI systems with already optimized predictive performance.

Model-agnostic methods can be used to generate explanations for predictions resulting from any trained AI model. These approaches often try to derive a less complex surrogate model for the trained ‘black-box’ model that needs to be explained [34]. Due to this, the explanations will be less faithful to the original model than the ones created using the model-specific approach.

Another important distinction is the scale of interpretation [46]. A local explainable AI method can provide explanations for specific prediction instances. Global methods explains how the model arrives at its predictions given any input.

For this project, the explainable AI method GNNExplainer [67] is used. A prediction obtained from any trained graph-based machine learning model serves as input of the GNNExplainer. Given the prediction in the form of a link prediction, node- or graph classification, the importance of node features and edges can be calculated. The GNNExplainer outputs a node or edge mask that shows which node features or edges in the GNN model are considered to be the most impactful for arriving at the given explanation. Considering the aforementioned classification of explainable AI methods, the GNNExplainer is a local model-agnostic approach.

2.9 Explainability

Explainability in the context of XAI can be described using the following definition:

An AI system is explainable if the task model is intrinsically interpretable (here the AI system is the task model) or if the non-interpretable task model is complemented with an interpretable and faithful explanation (here the AI system also contains a post-hoc explanation). [49]

In the given definition above, there is a distinction between AI systems in which the explanation is provided by the system itself and predictor models that are accompanied by a *post-hoc explanation*. For the latter, the explanation gives insight without having the knowledge about the mechanisms that are used in the predictor model. To relate to the classification of XAI methods, model-specific approaches are intrinsically interpretable while mode-agnostic methods provide *post-hoc explanations*.

For an AI system reaching explainability, both interpretability as well as fidelity are necessary [49]. Interpretability of an explanation is used as a measure of how understandable this explanation is for humans. The second measurement, fidelity, is about how accurately the explanation represents the behaviour of the model. A set of practical definitions has been created that can be utilized for measuring the level of explainability of an explanation.

As explainability can be split into interpretability and fidelity, these aspects can also be subdivided [49]. An explanation is considered to be interpretable if it is unambiguous and not too complex. The former is known as the clarity of the explanation while the latter is referred to as parsimony. Given these aspects, interpretability is about how a human can understand an explanation. Fidelity can be divided into completeness and soundness, representing the descriptive accuracy of an explanation. For completeness, the explanations need to provide sufficient information that tell how the model has acquired its output given the input. Explanations have soundness when it represents the inner workings of the model truthfully. In Figure 2.11, the hierarchy of all mentioned aspects of explainability is shown.

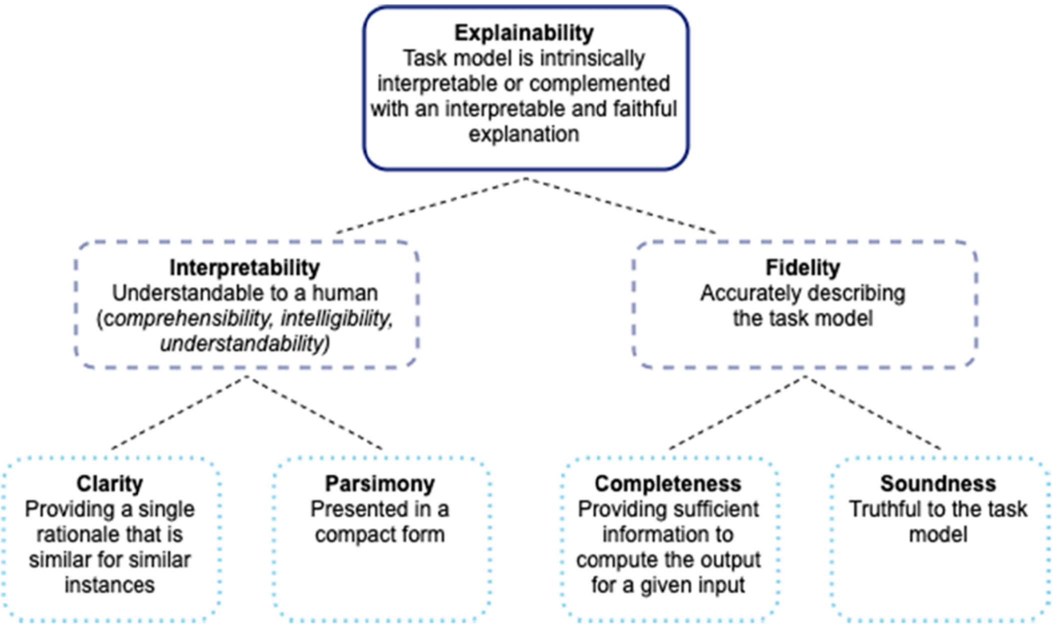

Explainability
Task model is intrinsically interpretable or complemented with an interpretable and faithful explanation
Interpretability
Understandable to a human
(comprehensibility, intelligibility, understandability)
Fidelity
Accurately describing the task model
Clarity
Providing a single rationale that is similar for similar instances
Parsimony
Presented in a compact form
Completeness
Providing sufficient information to compute the output for a given input
Soundness
Truthful to the task model

Figure 2.11: A hierarchical view of all aspects of explainability including their definitions taken from [49].

Since model-agnostic XAI methods compromise on the accuracy as they need to simplify the function learned by the task model to a less complex one, it can be expected that the explanations generated from these approaches will score less on the fidelity in comparison to model-specific XAI. This aspect of fidelity is however modified in order to create a more user-centric measurement on explainability. This means that with the fidelity aspect we try to measure in this project whether the explanations match how the user would explain the given prediction based on their knowledge and reasoning.

Chapter 3

Method

The method of this project enables the comparison of the explanations generated from the drug repurposing pipeline that uses as input a knowledge graph that complies to the conceptual structure from the XAI-DMD-DR project and the explanations from the pipeline using the restructured knowledge graph as input. To minimize the possibility that a difference in explainability between these sets of explanations is caused by a process that is not related to designing a new conceptual model, the drug repurposing pipeline has been kept the same. The overview of all steps performed to answer the research question is illustrated in Figure 3.1. The process is started by acquiring and aggregating data from multiple data sources resulting in a knowledge graph containing the necessary information for the drug repurposing task. This knowledge graph complies to the same conceptual structure as the input of the drug repurposing pipeline in the XAI-DMD-DR project. By applying the conceptual model design approach developed in this project, a restructured knowledge graph is created. Now, there are two knowledge graphs that can be used as input of the drug repurposing pipeline. Performing the processes in the pipeline for both input datasets, two sets of explanations are generated. To compare the explainability of these explanations, an explanation validation step has been added that will help us answer the research question of this project.

In the following method descriptions, the term *original* is used to refer to the variation of the drug repurposing pipeline that attempts to replicate that of the XAI-DMD-DR project [57]. The terms *restructured* or *new* are used for referring to the drug repurposing pipeline variation created during this project using another knowledge graph as input.

The implementation of the methods described in this chapter can be found at `https://github.com/rosazwart/XAIFO-ThesisProject`.

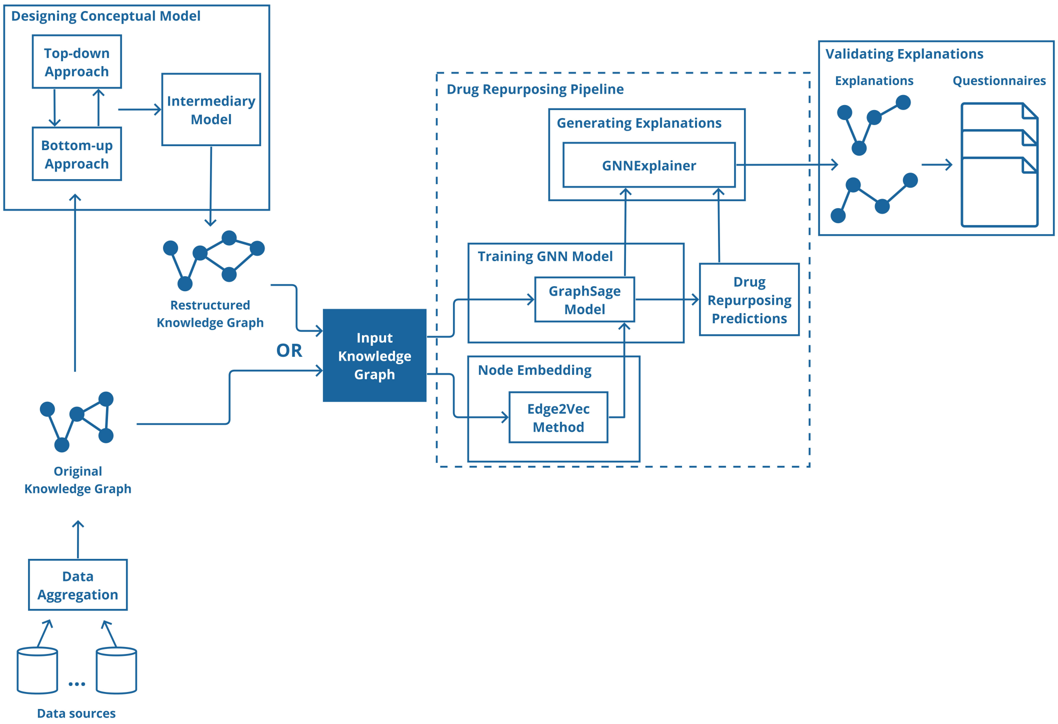


Figure 3.1: The schema that shows the summary of the method used in this project. The drug repurposing pipeline has the knowledge graph aligned to the conceptual model from the XAI-DMD-DR project as input or the knowledge graph that is restructured using the conceptual model design process of this project.

3.1 Data Aggregation

To facilitate performing the drug repurposing task, it is needed to collect different entities linked by various relations. For correlating symptoms related to DMD with drugs, genetic and phenotypic information as well as interactions between proteins and drug components are needed. This information is gathered by using multiple data sources being the Monarch Initiative data platform [59], DrugCentral [62] and the Therapeutic Target Database [69]. The same data sources have been consulted in the XAI-DMD-DR project. By combining the data from these sources and aligning it to our conceptual model, the knowledge graph that serves as the new input of the drug repurposing predictor was built.

The data aggregation process consists of multiple steps in order to collect and aggregate the information that populates the knowledge graph from multiple data sources. A global overview of this method is displayed in Figure 3.2.

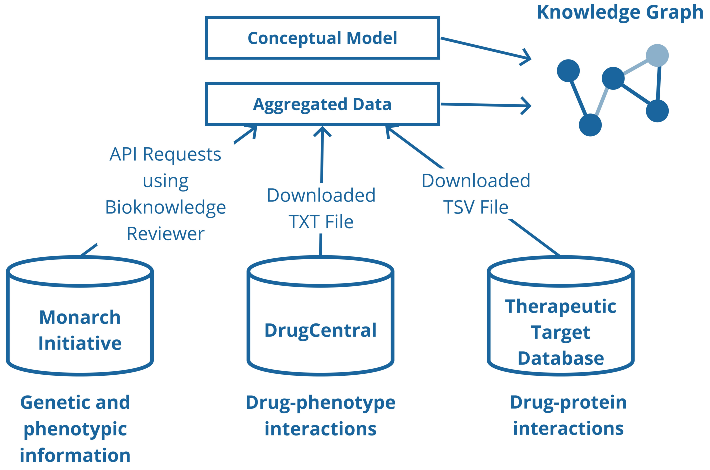

Knowledge Graph
Conceptual Model
Aggregated Data
API Requests using Bioknowledge Reviewer
Downloaded TXT File
Downloaded TSV File
Monarch Initiative
DrugCentral
Therapeutic Target Database
Genetic and phenotypic information
Drug-phenotype interactions
Drug-protein interactions

Figure 3.2: The schema that shows the collection of datasets used to build the knowledge graph. Three data sources are utilized being the Monarch Initiative data platform (`https://monarchinitiative.org/`), DrugCentral (`https://drugcentral.org/`) and Therapeutic Target Database (`https://db.idrblab.net/ttd/`). For genetic and phenotypic information, data is fetched using the API service of Monarch Initiative. To acquire the relevant information for the drug repurposing task, the already existing tool known as Bioknowledge Reviewer is applied. For drug-phenotype interactions, data is collected from DrugCentral. From TTD, drug-protein interactions are acquired. The data instances from all sources are aggregated such that a single knowledge graph is created that can be used in the next steps.

3.1.1 Bioknowledge Reviewer and Monarch Initiative

The Monarch Initiative provides an open-source data platform from which information can be retrieved about genes, variants, genotypes, phenotypes and diseases. Many widely used ontologies have been integrated into this platform which enables connecting the data from many different data sources into one large knowledge graph. This allows complex querying throughout various biological concepts and semantic relations. Genotype-phenotype associations are collected from a dozen of data sources covering over 100 species and originate from basic as well as clinical research [59].

For fetching the relevant data from the Monarch Initiative data platform used in the original knowledge graph, we reused the method from the XAI-DMD-DR project. This fetcher is part of the Bioknowledge Reviewer, a tool that collects knowledge from multiple sources and aggregates it to create a knowledge graph [56]. In this case, only the component of Bioknowledge Reviewer is used that extracts information from Monarch Initiative.

For the tool to extract data from Monarch Initiative, seeds must be provided in the form of identifiers. These identifiers are formatted differently for each category of entity. As information needs to be collected related to the rare disease DMD, the seeds given to the tool are MONDO:0010679 and HGNC:2928.

The first identifier represents the disease Duchenne muscular dystrophy and the second refers to the gene *DMD*. Based on the prefixes of their identifiers the latter entity is defined in the Mondo Disease Ontology (Mondo) [63]. The identifier prefix of the gene *DMD* indicates that the entity is recognized by the HUGO Gene Nomenclature Committee (HGNC) as an identified human gene approved by experts [58].

The Monarch Initiative data platform offers an API service that allows users to query its knowledge graph. Due to the data being represented in a single large knowledge graph, users can query and fetch entities of a particular category connected by specific relations. Given the seeds that serve as the initial nodes in the knowledge graph used for the drug repurposing pipeline, the Bioknowledge Reviewer fetches their first-order neighbouring nodes. From these neighbours, the orthologs of and the phenotypes associated to one of the seeds are collected and added to our knowledge graph. This step is repeated by obtaining the orthologs and associated phenotypes of these newly added nodes. Lastly, all first-order neighbours are collected given the seeds and nodes obtained by the previous steps.

Given the first steps of the Monarch Initiative fetcher, the focus is on building the base of our knowledge graph on orthologs[1] related to the gene *DMD*.

Fetching data used for building the restructured knowledge graph follows the same principles from the XAI-DMD-DR project for gathering relevant information from the Monarch Initiative data platform. However, some changes have been made to categorize the retrieved nodes. In the Bioknowledge Reviewer, the class of an entity is determined by the prefix of its identifier. Based on this method, it needs to be presumed that the prefix is exclusively used for a specific category. Inspecting the API service of Monarch Initiative, the responses include the class of the fetched entities. The edited version of the Bioknowledge Reviewer performed for building the restructured knowledge graph uses this information instead for categorizing the obtained nodes in order to stay true to the information provision of Monarch Initiative itself.

In this project, version 2.0 of the Bioknowledge Reviewer is used and can be found at `https://github.com/NuriaQueralt/bioknowledge-reviewer/blob/master/bioknowledge\_reviewer/monarch.py`.

The original knowledge graph is populated by fetching information from the Monarch Initiative on the 20th of February 2023. For the restructured knowledge graph, information has been fetched on the 8th of March 2023.

[1] A gene is orthologous to another gene when both genes are found in different species and are derived from a common ancestral gene. Genes that are orthologous to each other often have a similar function. These associations are of great value for the drug repurposing model when predicting new drug candidates.

3.1.2 DrugCentral

The open-access online drug compendium DrugCentral provides pharmaceutical information that is collected from different online public resources. Alongside the collection from multiple data sources, information also originates from manual curation of literature [62].

DrugCentral offers downloadable datasets found at `https://drugcentral.org/download`. For our knowledge graphs, the drug-target interactions are retrieved in a text file. It is worth noting that the download page does not provide the date on which the content of the file has been updated. Based on the modified date of the downloaded file itself, we can infer that the dataset originates from the 29th of October 2021. Each line in the document contains the name of the drug and the disease for which the drug is known to get prescribed. The diseases are only described using its name or short description, lacking identifiers. Therefore we mapped these labels to an identifier of the disease. In the XAI-DMD-DR project this is done by using the System for Ontology-based Re-coding and Technical Annotation (SORTA) [54]. This tool returns the most probable term from the Human Phenotype Ontology (HPO) [40] for each given disease label. Now, the entries can be associated with a phenotype represented by an identifier. Using a threshold for the scores of the matches, the obtained phenotypes can be aggregated with the phenotype nodes fetched from the Monarch Initiative data platform.

We used the described method for adding the drug-phenotype interactions to both the original and restructured knowledge graph.

3.1.3 Therapeutic Target Database

The Therapeutic Target Database (TTD) is a database that provides known drug targets. These targets can be proteins, pathways or diseases.

On the download page found at `https://db.idrblab.net/ttd/full-data-download` a tab-separated values file can be retrieved. Again, the page does not inform the user about the date of the most recent update of the downloaded file. Unlike the file from DrugCentral, the file from TTD only shows the date on which the user downloaded the dataset. The entries show the drug and the protein it is targeting. Each entry also shows the gene that produces this protein. The targets are identified with accession numbers used by the Uniprot Knowledgebase (UnitProtKB). In order to aggregate the targeted genes with the gene nodes retrieved from Monarch Initiative, these accession numbers have been mapped to the identifiers that are assigned to the same genes by the organism specific databases such as HGNC for the genes found in the human species. UniProt [13] provides a tool that enables the mapping from the UniProtKB gene identifiers to those used by the Monarch Initiative data platform.

The drug entities were aggregated with the already obtained drug nodes from the DrugCentral dataset using the names of the drugs. Ideally, this would need to have been done with identifiers instead. However, no method has been developed in the XAI-DMD-DR project and this project to enable this.

The described method is used for adding the drug-protein interactions in both the original and restructured knowledge graph. The file containing the interactions was accessed on the 15th of February 2023.

3.2 Conceptual Model Design

The conceptual model design process has been split into a top-down and a bottom-up approach visualized in Figure 3.3.

Step one of the process is an approach where the sources are limited by excluding the datasets and only referring to the known information about the overarching domain of the data. This can be considered as a top-down method as the expressed concepts and relations using a foundational ontology are derived from collected knowledge about the relevant domain. This first step leads to the creation of the *domain (reference) model/domain-based model.*

The second step is the bottom-up approach which entails that a model is aligned to the concepts and relations found in exclusively the dataset itself resulting in the *data-based model.* For this, the dataset is considered that results from solely fetching and linking the data entries coming from the Monarch Initiative, DrugCentral and TTD databases. This is the method used in the XAI-DMD-DR project during which no additional changes have been applied to the conceptual structure of the composed knowledge graph used as input of the drug repurposing predictor. This graph will be referred to as the *original knowledge graph* throughout the research project.

The *final conceptual model* is built by comparing and finding consensus between the *domain reference model* and the *data-based model.* This is referenced as the third step in Figure 3.3. Comparing the two draft models facilitates validation of the *data-based model* as we can compare its concepts and conceptual structures to those expressed in the *domain reference model.* This is especially useful when the dataset is composed of entries originating from sources that comply with different ontologies. This hybrid approach harmonizes [48] the heterogeneity of the dataset since a mapping will be yielded during comparison of the *domain-based* and *data-based model.* By aligning the dataset with the concepts and relations in the *final conceptual intermediary model*, a restructured dataset is acquired that will be used as the new input of the drug repurposing predictor. This dataset is referred to as the *restructured knowledge graph* in this project.

For all conceptual models built throughout the conceptual model design pro-

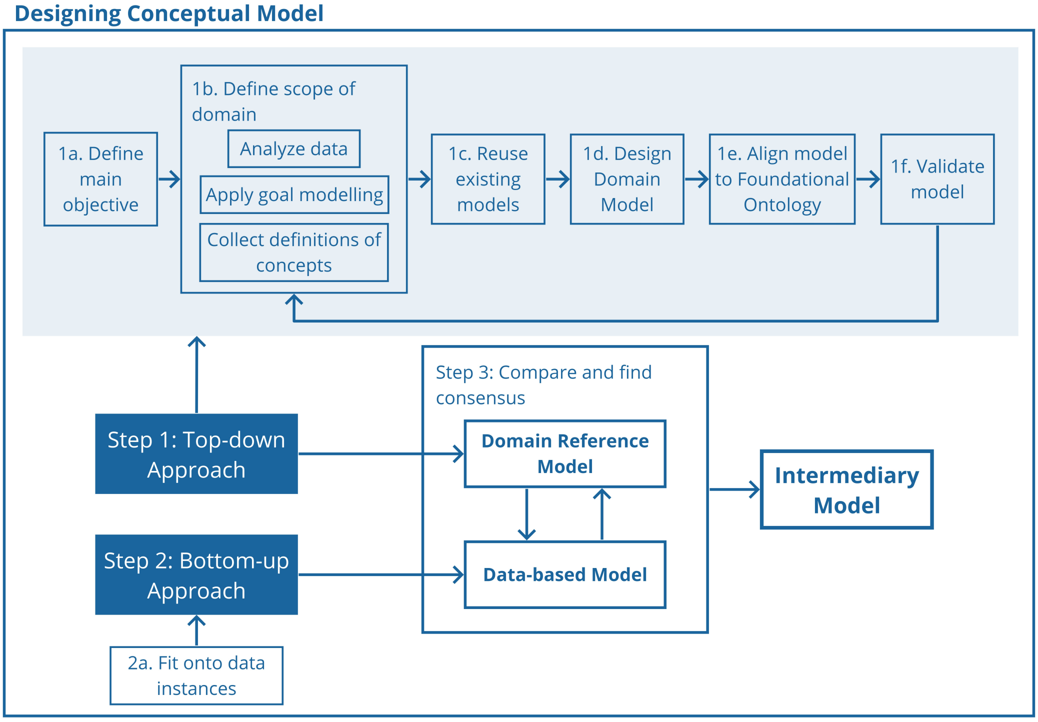

Designing Conceptual Model
1a. Define main objective
1b. Define scope of domain
Analyze data
Apply goal modelling
Collect definitions of concepts
1c. Reuse existing models
1d. Design Domain Model
1e. Align model to Foundational Ontology
1f. Validate model
Step 1: Top-down Approach
Step 2: Bottom-up Approach
2a. Fit onto data instances
Step 3: Compare and find consensus
Domain Reference Model
Data-based Model
Intermediary Model

Figure 3.3: The schema inspired by [48] that shows the hybrid approach towards creating a conceptual model to which the original dataset can be mapped. The steps include a top-down (step 1) and bottom-up approach (step 2) resulting in a *domain-based* and *data-based model*, respectively. The two draft models are compared and a consensus is found between them yielding a *final conceptual model* (step 3).

cess, the OntoUML plugin (version 0.5.3) for Visual Paradigm is used. This plugin can be accessed via `https://github.com/OntoUML/ontouml-vp-plugin`. Visual Paradigm found at `https://www.visual-paradigm.com/` is a software tool that can be used for many different modelling approaches and enables the addition of plugins to even further increase the variety of models that can be built.

3.2.1 Top-down Approach

Multiple steps need to be performed in order to build an accurate conceptual model representing the relevant domain. This model must also provide the concepts and relations that are sufficient for reaching the main objective. All steps of this design process are shown in Figure 3.4.

Defining the Main Objective

For step 1a in Figure 3.4, the main objective is defined. Since this project builds upon the XAI-DMD-DR project, the main objective of this previous project

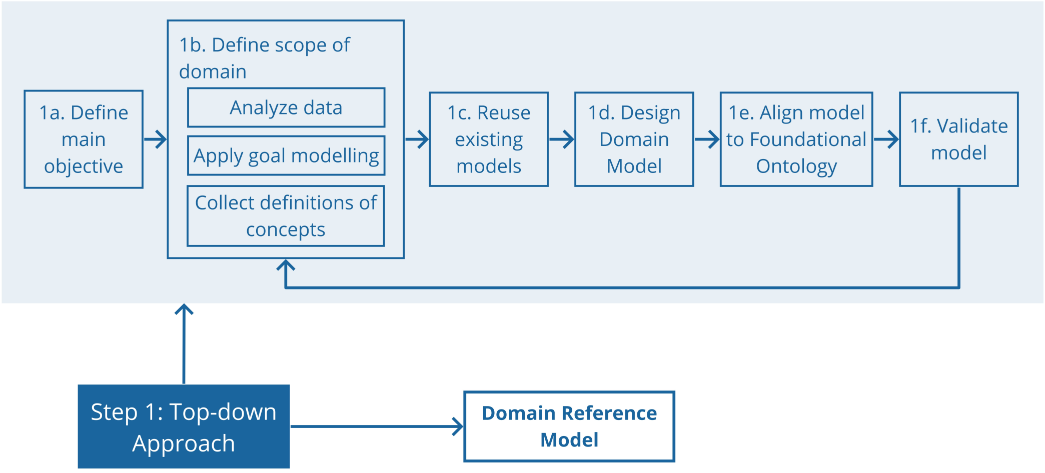

1a. Define main objective
1b. Define scope of domain
Analyze data
Apply goal modelling
Collect definitions of concepts
1c. Reuse existing models
1d. Design Domain Model
1e. Align model to Foundational Ontology
1f. Validate model
Step 1: Top-down Approach
Domain Reference Model

Figure 3.4: An excerpt of the diagram from Figure 3.3. It shows the needed steps for designing the *domain model*. Step 1a: The main objective is defined. Step 1b: By briefly analysing the dataset, collecting relevant definitions and applying goal modelling, the needed scope of the domain is identified and justified. Step 1c: Existing conceptual models are searched for that can be partially or completely reused in the *domain model*. Step 1d: the structure and content of the model is designed based on all acquired findings. Step 1e: To express the intended meanings contained in the model, the model is aligned to a foundational ontology. Step 1f: The last step is the validation of the accuracy of the model by domain-experts. In case of the model being rejected by the experts, the previous steps need to be reiterated while incorporating the received feedback.

is replicated which is “obtaining drug candidates that can treat the symptoms observed with the disease DMD based on their already known targets” [57].

Defining the Scope

The domain to be modelled can be extended into the smallest details. Thus, it should be cared for that the granularity of concepts and relations between them is defined. It is needed to balance between providing sufficient information and making it doable to capture all the required details. Thus, the scope of the domain needs to be defined which is step 1b in Figure 3.4.

The scope can be determined by briefly looking at the general structure of the available data in order to get an initial notion about the minimum granularity that can be achieved.

During this project, it was initially difficult for the consulted experts to understand the scope and purpose of the *domain model* due to their complexity. Using goal modelling, the scope can be identified and justified by splitting the main objective into multiple manageable subgoals. It also helped clearly showing how the scope of the domain represented in the conceptual model is established based on our main objective of the drug repurposing pipeline. The hierarchical goal model is included in Figure 3.5 and will be discussed in the next paragraphs.

As mentioned before, the main objective would be to find drug candidates for the disease DMD. In order to obtain these, drugs need to be investigated that are known to treat other diseases. This is done by training a GNN model that yields probability scores of a drug treating a DMD-related symptom. Ideally, for each drug candidate prediction an explanation can be generated that is rich and provides good explainability because it needs to support decision making.

Two subgoals (Fig. 3.5) have been described that are assumed to provide sufficient information for the XAI algorithm to predict new links between DMD or its associated phenotypes and drugs. Also, this information would enable the user to make sense out of the generated explanations as it will be constructed with a manageable but informative amount of concepts and relations.

The first subgoal would be to collect data such that it is possible to find associations between drugs that target gene products and genes linked to DMD. For obtaining this information, the mentioned Therapeutic Target Database and Monarch Initiative data platform are referred to, replicating the data fetching method of the previous project. The former database provides data instances that represent drugs being known to target certain gene products. The data platform Monarch offers genetic and phenotypic information that is helpful for finding various entities of different concepts that are associated in some way with DMD. The second subgoal states that data needs to be collected in order to be able to find associations between drugs that are used to treat certain diseases or symptoms and diseases or phenotypes that are associated with DMD. From DrugCentral, information is gathered that shows for what phenotypes or diseases certain drugs are used. Again, associations between phenotypes and DMD can be discovered with the genetic and phenotypic information gathered from Monarch Initiative.

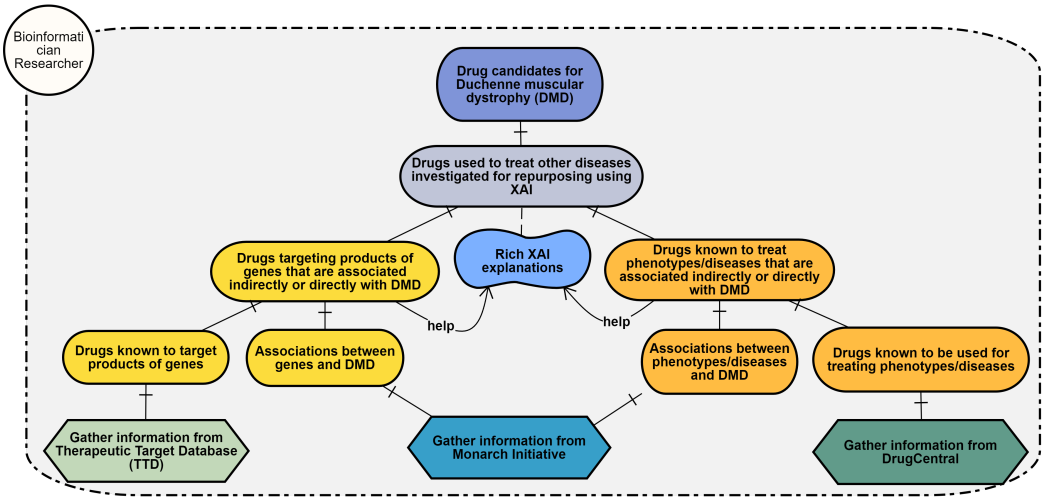

Bioinformatician Researcher
Drug candidates for Duchenne muscular dystrophy (DMD)
Drugs used to treat other diseases investigated for repurposing using XAI
Rich XAI explanations
Drugs targeting products of genes that are associated indirectly or directly with DMD
Drugs known to treat phenotypes/diseases that are associated indirectly or directly with DMD
help
help
Drugs known to target products of genes
Associations between genes and DMD
Associations between phenotypes/diseases and DMD
Drugs known to be used for treating phenotypes/diseases
Gather information from Therapeutic Target Database (TTD)
Gather information from Monarch Initiative
Gather information from DrugCentral

Figure 3.5: Goal model using the iStar Framework [15]. The rounded shapes are *goals*, rectangles are *resources*, hexagons are *tasks* and the cloud-shaped icons represent *quality*. For the different relations between these elements, there is the *AND-relation* specified with a dash through the edge. The *contribution links* are shown with a regular arrow together with an edge label. Lastly, the edge with a dot shows the *needed-by relation*. This goal model specifies the main objective and its breakdown into multiple high level sub goals. Given this model, the main goal is to determine drug candidates that can potentially treat symptoms of DMD. This objective is reached in turn by using explainable AI to predict these drug candidates. The goal model also shows how predictions and rich explanations of the explainable AI algorithm are achieved by requiring as input data associations between drugs, genes, phenotypes and DMD gathered from multiple sources. A higher quality version of this goal model can be found at `https://github.com/rosazwart/XAIFO-ThesisProject/tree/main/images`.

Considering the mentioned subgoals will not help with specifying the needed scope of the domain yet but it can be seen as a useful foundation upon which new subgoals can be built again. The first subgoal is split into more specific goals shown in Figure 3.6 in order to show how associations between genes and the disease DMD can possibly be found taking into account the information that can be gathered from the Monarch Initiative data platform. The information can come in the form of entities that belong to semantic groups representing genes, diseases, biological processes, anatomical structures, phenotypes, variants and genotypes. DMD has direct associations with genes, as they can be causal or correlated to this disease. There might also exist associations between DMD and genes that are not explicitly expressed in the data set. It can be hypothesized that these indirect associations are considered as informative paths for the prediction of new links between DMD and drugs by the XAI:

- Genes can interact with other genes, as the product of one gene can have an effect on another gene due to regulatory functionalities or the products of both genes are present in the same biological mechanism such as a pathway. Genes can thus interact with the genes that are associated with DMD which relates genes to DMD while they were initially not connected with the disease.

- Genes found in other species can have an orthologous relation with a gene found in the human genome. This means that the genes are evolutionary related to each other, as they are derived from the same gene found in the ancestor of both species [51]. It often occurs that genes orthologous to each other fulfil the same functionalities. Genes that are orthologous to causal or correlated genes of DMD, can possibly be associations useful for the predictions aimed for.
- The disease DMD can be involved in biological processes. It can be speculated that genes expressing in the same biological processes are associated in some way with DMD.
- Based on all variants associated with DMD, the genes can be inferred of which the variants are known to be an allele. These are paths that might be informative for predicting as it highlights possible new associations between DMD and genes.
- Lastly, links between genes and DMD can be inferred from associations between this disease and phenotypes or other diseases. As there is also information about which variants or genes cause certain phenotypes or diseases, it is possible to find new associations between DMD and genes following these information paths.

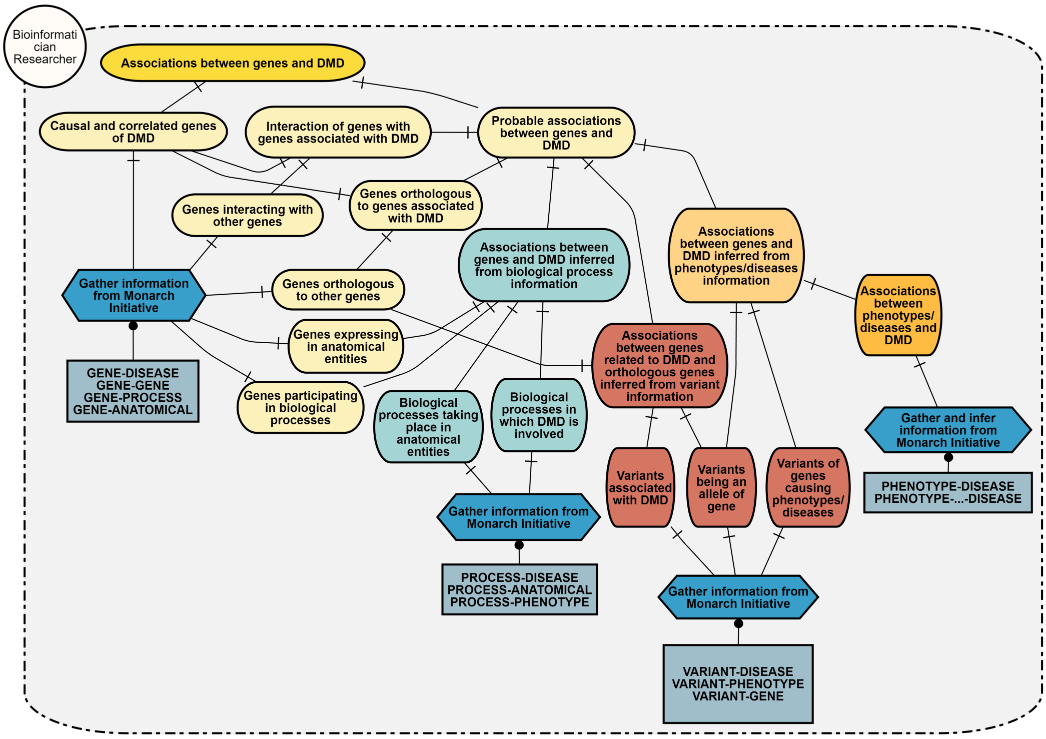

Bioinformati
cian
Researcher
Associations between genes and DMD
Causal and correlated genes of DMD
Interaction of genes with genes associated with DMD
Probable associations between genes and DMD
Genes interacting with other genes
Genes orthologous to genes associated with DMD
Associations between genes and DMD inferred from biological process information
Associations between genes and DMD inferred from phenotypes/diseases information
Gather information from Monarch Initiative
Genes orthologous to other genes
Associations between phenotypes/ diseases and DMD
GENE-DISEASE
GENE-GENE
GENE-PROCESS
GENE-ANATOMICAL
Genes expressing in anatomical entities
Associations between genes related to DMD and orthologous genes inferred from variant information
Genes participating in biological processes
Biological processes taking place in anatomical entities
Biological processes in which DMD is involved
Gather and infer information from Monarch Initiative
PHENOTYPE-DISEASE
PHENOTYPE-...-DISEASE
Variants associated with DMD
Variants being an allele of gene
Variants of genes causing phenotypes/ diseases
Gather information from Monarch Initiative
PROCESS-DISEASE
PROCESS-ANATOMICAL
PROCESS-PHENOTYPE
Gather information from Monarch Initiative
VARIANT-DISEASE
VARIANT-PHENOTYPE
VARIANT-GENE

Figure 3.6: Goal model specifying all possibly useful information paths for associating genetic data entities with the disease entity DMD that can be used by the XAI algorithm in order to yield accurate drug repurposing predictions generating rich explanations. A higher quality version of this goal model can be found at `https://github.com/rosazwart/XAIFO-ThesisProject/tree/main/images`.

Based on the data available from the Monarch Initiative dataset, phenotypes can be associated with diseases. This is specified in the model as the second subgoal that needs to be achieved in order to investigate drugs that are known to treat other diseases (Fig. 3.5). This subgoal has been extended into multiple goals and tasks shown in Figure 3.7. In addition to direct associations, associations between DMD and phenotypes or other diseases may also be inferred indirectly in the following ways:

- The group of genes associated directly or indirectly with DMD can be linked to the information about what phenotypes are caused by what genes. This also includes for example genes orthologous to genes associated with DMD.
- It can be considered that it is also possible to find links between DMD and phenotypes by looking at all phenotypes describing a biological process in which the disease DMD is involved. These connections alone are weak, but might help supporting associations from other inferences.
- Informative paths can also be found looking at variant information.

There are direct connections between DMD and variants. The dataset also provides which phenotypes are caused by which variants when this is known.

- It is possible that a single variant does not cause any phenotype interesting for forming the drug repurposing predictions. However, combining this variant with variants of other genes does cause relevant phenotypes. This is why genotypic information should be included in the training set of the GNN. It might be possible to relate sets of variants associated with DMD to genotypes. From these connections, phenotypes can be found that are known to be caused by these genotypes. Specifically for the case of DMD, this association path might be less informative due to the fact that DMD is known as a disease caused by a mutation in a single gene.
- Lastly, other diseases can be related to DMD when there is a large overlap of associated phenotypes between them.

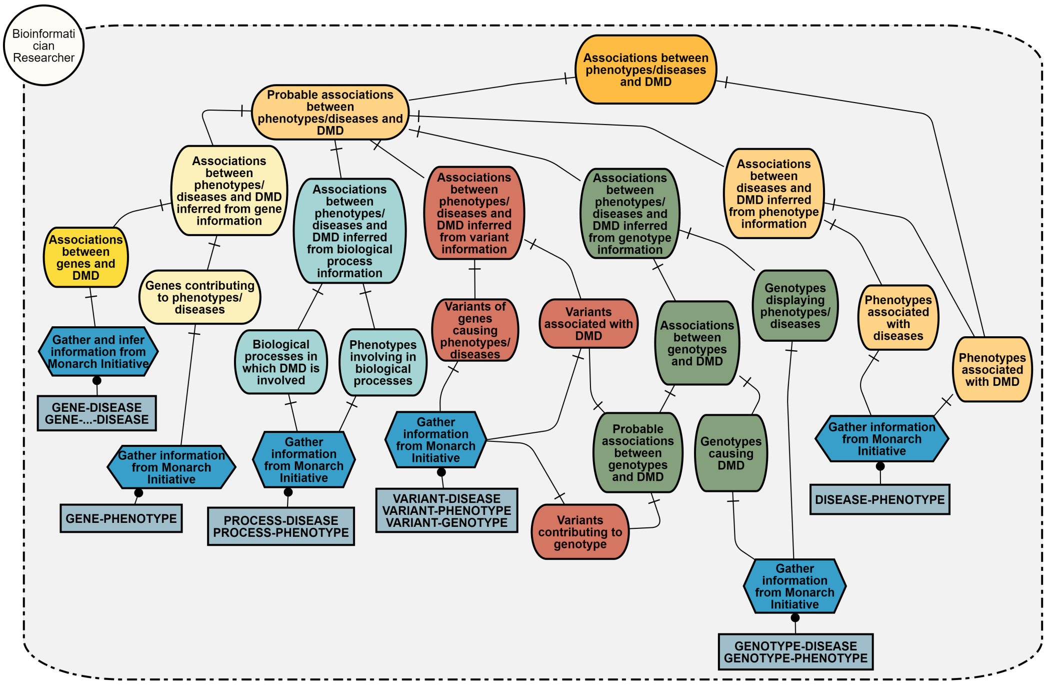

Bioinformatician Researcher
Associations between phenotypes/diseases and DMD
Probable associations between phenotypes/diseases and DMD
Associations between phenotypes/diseases and DMD inferred from gene information
Associations between phenotypes/diseases and DMD inferred from biological process information
Associations between phenotypes/diseases and DMD inferred from variant information
Associations between phenotypes/diseases and DMD inferred from genotype information
Associations between diseases and DMD inferred from phenotype information
Associations between genes and DMD
Genes contributing to phenotypes/diseases
Variants of genes causing phenotypes/diseases
Variants associated with DMD
Associations between genotypes and DMD
Genotypes displaying phenotypes/diseases
Phenotypes associated with diseases
Phenotypes associated with DMD
Gather and infer information from Monarch Initiative
Biological processes in which DMD is involved
Phenotypes involving in biological processes
GENE-DISEASE GENE-...-DISEASE
Gather information from Monarch Initiative
Gather information from Monarch Initiative
Gather information from Monarch Initiative
Probable associations between genotypes and DMD
Genotypes causing DMD
Gather information from Monarch Initiative
GENE-PHENOTYPE
PROCESS-DISEASE PROCESS-PHENOTYPE
VARIANT-DISEASE VARIANT-PHENOTYPE VARIANT-GENOTYPE
Variants contributing to genotype
DISEASE-PHENOTYPE
Gather information from Monarch Initiative
GENOTYPE-DISEASE GENOTYPE-PHENOTYPE

Figure 3.7: Goal model specifying all possibly useful information paths for associating phenotypic data entities with the disease entity DMD that can be used by the XAI in order to yield accurate drug repurposing predictions generating rich explanations. A higher quality version of this goal model can be found at `https://github.com/rosazwart/XAIFO-ThesisProject/tree/main/images`.

It is not possible to say that all these information paths are used by the GNN model to produce its predictions. However, these associations are hypothesized to be potentially informative for the model to use for arriving at its predictions.

This might be in the form of utilizing single conceptual information paths or using associations from many different inferences mentioned in the goal models (Fig. 3.6 and 3.7).

Collecting Definitions of Concepts

Definitions of the relevant concepts need to be found that are considered to have a high consensus among the experts of the domain. For instance, definitions of basic concepts can be collected from well-known textbooks that cover the relevant domain. There exist tools that use heuristic and machine learning approaches in order to extract excerpts of definitions for a given concept originating from various published literature in the forms of articles, reviews and books. This functionality is for example provided by Elsevier's platform of peer-reviewed scholarly literature known as ScienceDirect.

Reusing Existing Models

Applying a foundational ontology to the designing process of the *domain model* facilitates using already existing models aligned to the same foundational ontology. Reusing parts of established models increases the quality of the newly built model while benefiting from the reusability feature provided by the application of a foundational ontology. From literature studies a model (Fig. 3.8) is found that covers a part of the relevant domain [50] and has been aligned to the foundational ontology Unified Foundational Ontology (UFO). This model expressing variation in the human genome shows concepts that are not covered by the defined scope, such as the sequence of nucleotides forming an allele. However, the part that expresses that an allele is either an allelic reference or variant can be included in the *domain reference model*. Based on the defined scope, the part of the model that states that multiple alleles represent a gene as a group can be considered to be added to the conceptual model that we are developing.

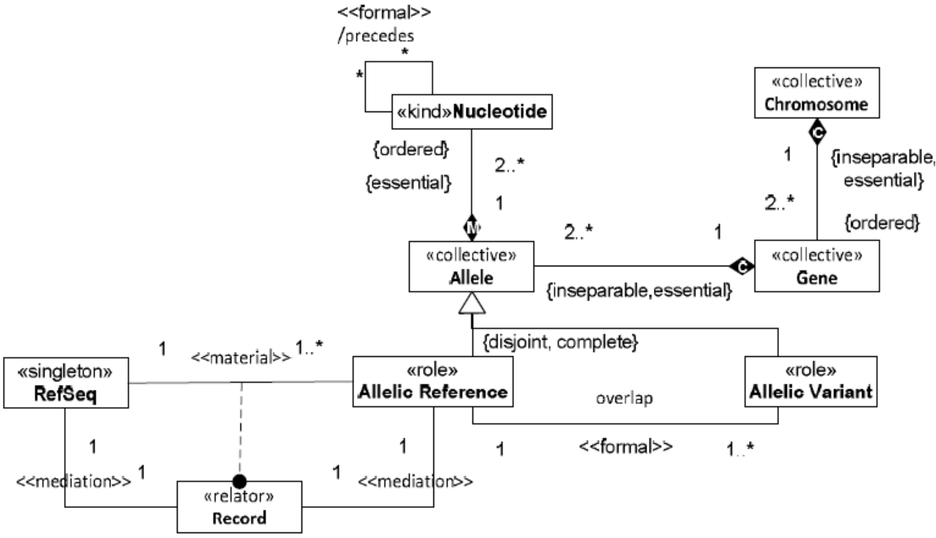


Figure 3.8: A conceptual model that represents gene variation expressed with OntoUML [50]. The concepts of an allele being either an allelic reference or a variant and a gene represented by multiple alleles have been added to our *domain reference model*.

Design Model

Considering the previous steps, information has been acquired that is needed for designing the model itself. All found terms and relations that can exist between them are included in the model. After completing this design step, the model is aligned to a foundational ontology.

Aligning Model to a Foundational Ontology

The model is aligned to a foundational ontology to improve the reality adherence of its conceptual structure. To achieve this, we utilize OntoUML that helps to define concept structures that comply with the categorization and axiomization of UFO [29]. OntoUML facilitates building a model since it enables expressing a model in a concise way using well-defined domain-independent conceptual elements. This domain-independent foundation upon which models can be built, increases understandability of the concepts despite its domain-specific content.

The complete overview of the *domain model* can be seen in Figure A.1. For better understandability, the view is also split into multiple modules to explain the content of the model.

The model expresses the following statements in the partial view provided in Figure 3.9:

- An *allele* is defined as being a DNA sequence found at a specific site on a chromosome implying that an allele is some variation of a certain gene.

A collection of *alleles* represents a *gene*.

- A *gene* is found in one *species*.
- An *allele* is a *reference allele* when it has been recorded as the reference. Otherwise, an *allele* is a *variant allele*.
- A *reference allele* and a *variant allele* differ because of one or multiple *sequence variations*.
- A *gene* of one *species* can be orthologous to a gene of another species due to a *speciation divergence* that is known to have happened at some point in time. This is called an *ortholog gene*.
- A *genotype* specifies a genome with a set of variants along the genetic background. A *genotype* can be specified by this collection of *variant alleles* that represent different *genes*.

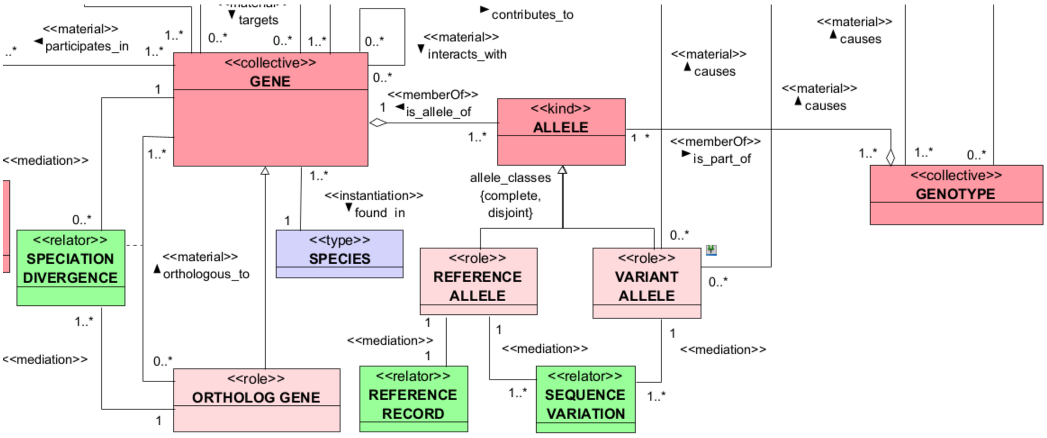


Figure 3.9: A partial view of the *domain model* expressed with OntoUML. This part of the model consists of the concepts *genes*, *orthologous genes*, *gene variants* and *genotype*.

In Figure 3.10, the part of the model is shown containing the concepts *biological process*, *gene* and *anatomical structure*:

- A *biological process* is any process that occurs in a living organism. Such a process can exist of multiple interactions between molecules. Thus, a *biological process* can be either a *complex* or *atomic process*. The former indicates any process associated with multiple subprocesses while the latter represents the lowest level of processes that are not known to have subprocesses related to them.
- The *biological processes* are observed to take place in one or more *anatomical entities*. The definition of an *anatomical entity* is taken from the Common Anatomical Reference Ontology [30] defining this concept as

being entities that are a part of a cellular organism and have a granularity level above protein complexes.

- A *gene* can participate at some level in *biological processes*. Its product can have different functionalities during a process, such as serving as a regulator.
- *Genes* are known to express in one or more *anatomical entities*.

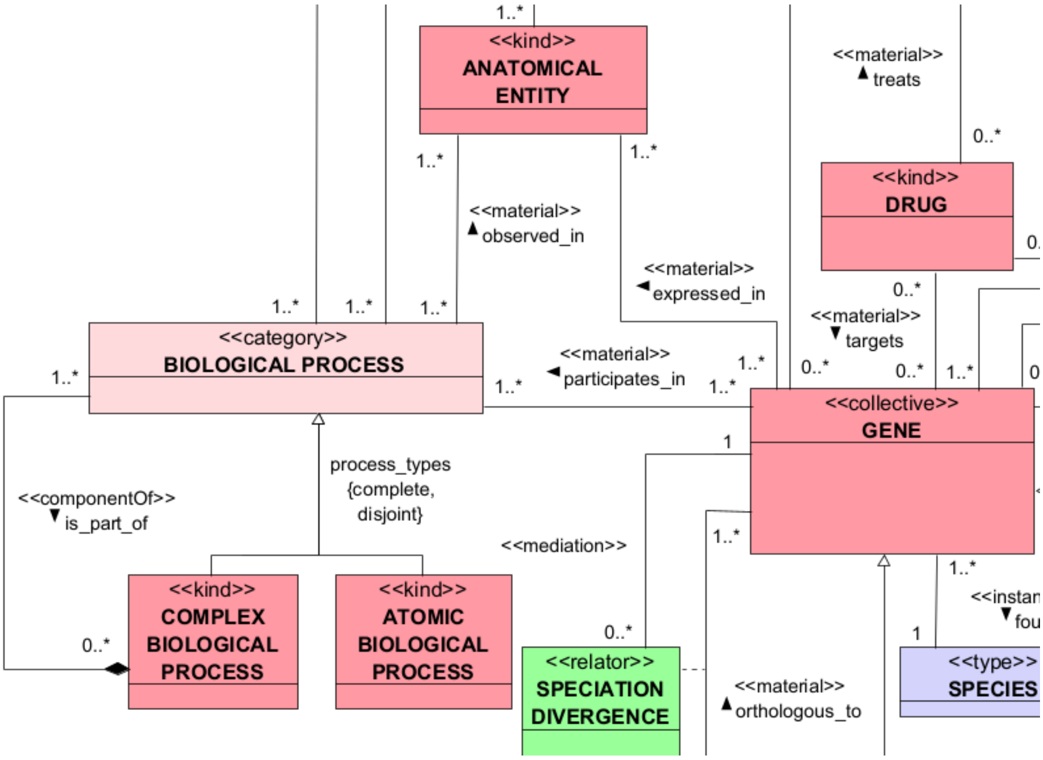


Figure 3.10: A partial view of the *domain model* expressed with OntoUML. This part of the model consists of the concepts *biological process*, *anatomical structure* and *gene*.

In the last part of the model shown in Figure 3.11, the concepts *phenotype*, *disease* and *drug* are introduced:

- A *phenotype* can be defined as "the observable characteristic of an organism which arises from complex interactions between its *genotype* and its environment" [22]. Considering the current scope of the dataset, environmental data entries are not available. This means that the focus lies on the genetic influence on occurrences of *phenotypes*. This influence can originate from several factors. A *gene* can contribute to one or more *phenotypes*. This might be the case due to its product playing an important role in a *biological process* that is involved in enabling the relevant *phenotype*. A specific *variant* of a *gene* can also be known to cause specific *phenotypes*. A change in the protein structure caused by the mutation in the *gene* can result in a *phenotype* that is different from the *phenotype* that would occur in the absence of this gene variance. A *phenotype* might be the result of the presence of a set of *variants* of different *genes* which is expressed by stating that a *genotype* is observed to cause one or more *phenotypes*.

- *Phenotypes* can be classified into *phenotypes* that describe a *physiological process* or an *anatomical aspect* [22].
- The concept *disease* is not very informative when it comes to connecting one *disease* to another. A *disease* can be specified by the set of *phenotypes* associated to them which represents its known clinical picture.
- A *disease* can be further specified by including a connection to one or more *biological processes* since *diseases* might be known to affect some specific processes.
- It can be stated that a *gene* contributes to a *disease*. More specifically, it could be said that a *gene variant* causes a *disease*. The presence of a combination of multiple *variants* might cause a *disease* while only having one of these *variants* in a genome does not give rise to the same disease. This substantiates the need of connections that indicate a possible causal relation between *genotypes* and *diseases*.
- A *drug* needs to treat at least one *disease* in order to consider the substance to be a *drug*. *Drugs* can be known to specifically have an effect on one or more *phenotypes*. Also, the substance might target specific products of *genes* which is represented by the connection between the concepts *drug* and *gene*.

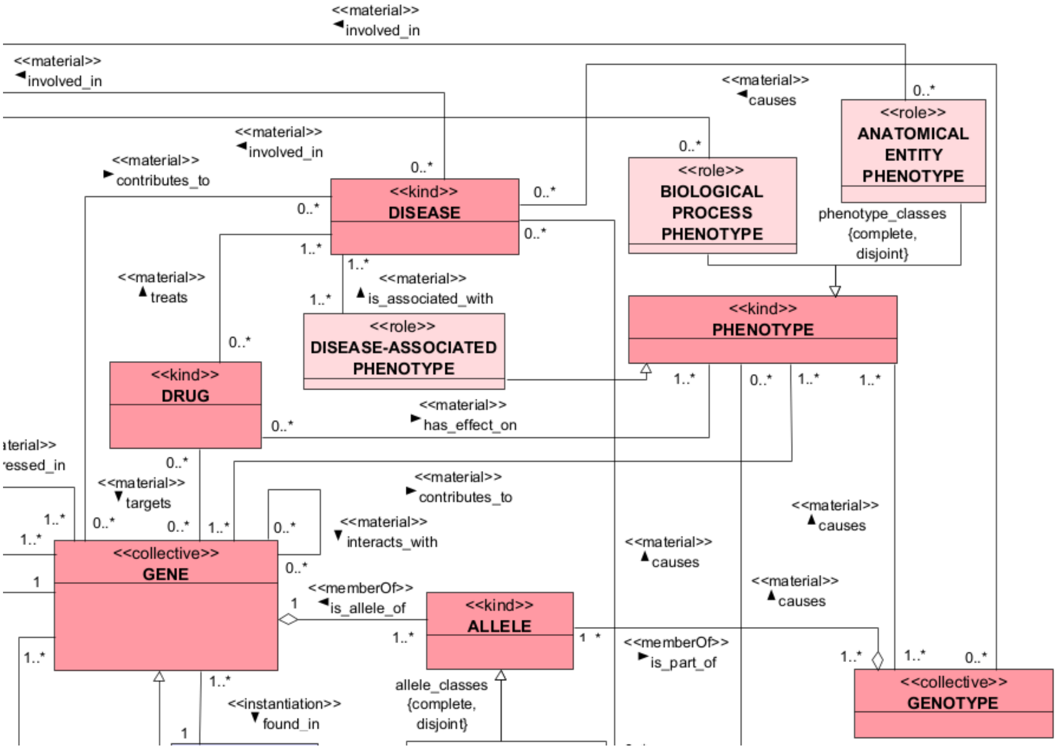


Figure 3.11: A partial view of the *domain model* expressed with OntoUML. This part of the model consists of the concepts *gene*, *variants*, *genotype*, *phenotype*, *disease* and *drug*.

Model Validation

A discussion has been held with multiple members of the Leiden University Medical Center (LUMC) Biosemantics group. This group consists of people who are researchers on rare diseases or experts in ontology modelling. Many members also work extensively with medical and genetic data. For example, one member contributes to the Leiden Open-source Variation Database (LOVD). The LOVD is an open-source tool for collecting, displaying and curating genomic variants and phenotypes [20]. Due to this, they have been considered experts suitable for giving feedback about our designed *domain model*.

The implementation of presenting the scope of the *domain model* in a goal model format did result in a better understanding of the presented *domain model*, resulting in experts being able to give feedback that is relevant to the objective that needs to be achieved using this model.

The design steps had to be reiterated in order to implement the received feedback. Finally, this resulted in a validated *domain model* that can be compared with the *data-based model* in order to acquire the *final conceptual model*.

3.2.2 Bottom-up Approach

For the second step of the hybrid approach, the model is built using the available data instances as reference for the concepts and relations that need to be included in the *data-based model*. In order to do this, the overall structure of the set of data instances needs to be analysed. The original dataset contains a total of eight concepts. In Figure 3.12 a graph is included that shows the original structure of this dataset. In this graph, the edges indicate that the concepts are connected to each other by one or more relations. This is the state of the dataset before aligning the data to the *final conceptual model*. The specifications of which relations are linking which concepts are shown in Table C.1 in the appendix.

All concepts and relations found in the original dataset have been included in the conceptual model shown in Figure A.2 in the appendix. Again, OntoUML has been applied to express the structure.

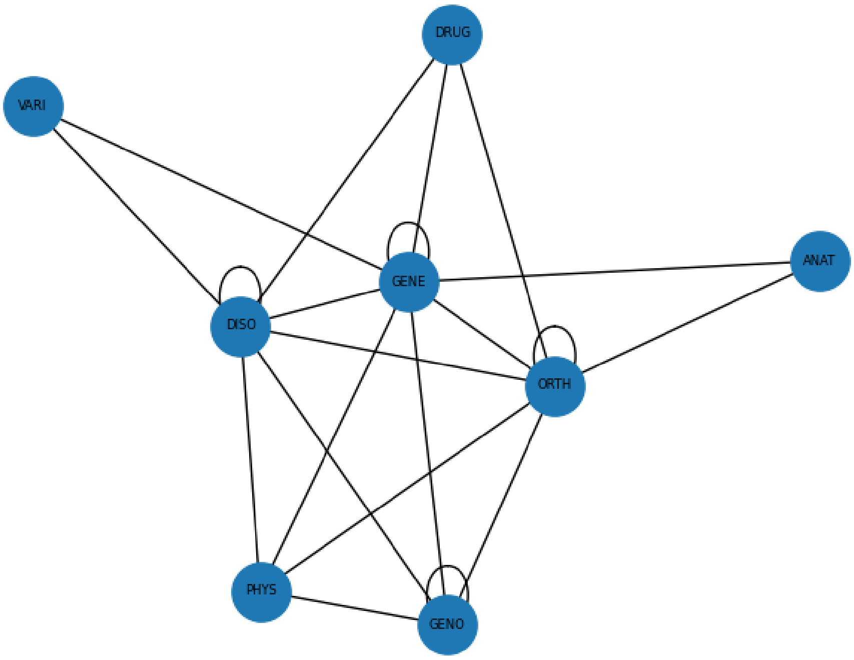


Figure 3.12: A graph in which each edge represents one or more relations that exist between the two connected concepts present in the original dataset acquired by using the data fetch scripts from the previous project [57]. In this figure *VARI* stands for variants, *DISO* for diseases, *ORTH* for orthologs, *ANAT* for anatomical structures, *PHYS* for physiological functions and *GENO* for genotypes.

3.2.3 Aligning Domain-based Model with Data-based Model

The *domain model* (Fig. A.1) and *data-based model* (Fig. A.2) are compared to each other yielding the final middle-level conceptual model shown in Figure A.3.

The decisions that have been made during the alignment are based on suggestions from domain experts and literature, avoiding as much decision-making by ourselves. The decisions are based on the goal to avoid over-complexity, as we hypothesise that this may introduce too much noise and thus negatively affect predictive and explanatory performance.

In this section, we often refer to the relations and their definitions. For readability, the references are excluded here when citing the definitions. Instead, the identifiers and annotations of these relations have been collected in an overview in Table B.1 showing their ontology source.

Concepts

There are some differences between the concepts present in the *data-based model* and the *domain model*:

- In the *data-based model*, orthologous genes are considered a different concept from the concept gene. In the *domain model*, it is highlighted that an ortholog is not different from a gene except for the occurrence of at least one link to another gene entity that indicates an orthologous relationship between them. The emphasis of an ortholog being a gene is also included in the *final conceptual model* by removing ortholog being a separate concept from gene. In the *final model*, an entity of concept gene can have as subtype ortholog when it is known to have an orthology relationship with another gene.
- The concept phenotype cannot be found in the *data-based model*, because all phenotype entities are considered to belong to the concept disease. In the *domain model*, a disease is seen as an entity that can be associated with a collection of phenotypes. Some disease-associated phenotypes might play a more prominent role than other phenotypes in the clinical picture of the disease which implies that a disease is conceptually seen as a functional complex. This will also be emphasized in the *final model* by considering the disease as the OntoUML class stereotype *Kind*. Also, the concept phenotype is added to the model and can have a relation with a disease entity when it is considered to belong in the clinical picture of that disease.
- Based on the connection graph in Figure 3.12, the anatomical entity concept has associations with very few other concepts. In the *domain*

model, it is expressed that phenotypes, diseases, biological processes and genes can be linked in some informative way with an anatomical entity. However, the *data-based model* shows that there are only associations between anatomical entities and genes. Since this concept already carries a low granularity as a very large number of processes can take place in an anatomical entity, the presence of these entities in the dataset will not help with inferences about for example gene similarity, Thus, this concept has been excluded in the *final conceptual model*.

- In the *domain model*, the species in which genes are found is included while this is not the case in the *data-based model*. In the original dataset, it is already specified by categorizing the entity under ortholog instead of gene when this gene is found in a species other than the human species. In the *final conceptual model*, orthologs are subtypes of the concept gene. Because of this, it is needed to specify in what species the gene is found which is done by including the concept species to the *final model*.

Relations

Looking at the *data-based model*, it can be noticed that there is a large number of relations existing between the same pairs of semantic groups while this is not the case in the *domain model*. Some of these relations refer to information that will not add anything that improves the drug repurposing predictions considering the concepts available in the dataset. On the contrary, by including relations that are very similar to other relations given the limiting granularity that can be reached, it would only cause more noise to the dataset. Thus, it is investigated whether the relations in the *data-based model* hold the same level of information considering the scope defined by the goal model. A relation will be levelled to its ancestor when it is concluded that for the given scope the occurrence of the descendant relation in the dataset will not yield additional information.

The relations present in the dataset are defined in different ontologies being the OBO Relations Ontology [52] and the GENO Ontology [12]. The Ontology Lookup Service [38] provides a querier that is able to collect all ancestors and descendants of given relation entities from various ontology sources. This querier has been implemented such that it could be used for deciding whether relations present in the dataset can be merged with other relations.

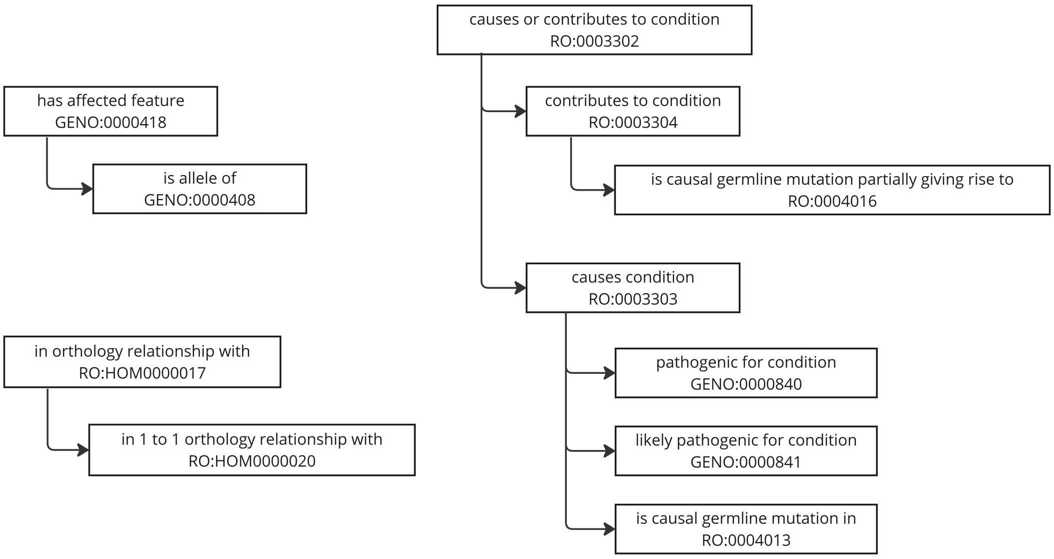


Figure 3.13: The hierarchy between relations in the original dataset based on the OBO Relations Ontology and the Genotype Ontology. All relations except for *causes or contributes to condition* can be found between some concept entities in the dataset. The arrows point towards the child of the relation.

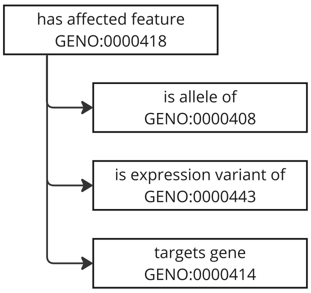


Figure 3.14: The relations that are children of the relation has *affected feature* based on the Genotype Ontology.

Figure 3.13 shows the hierarchy present in the original dataset. Based on the scope taken into account, it can be decided whether some relations need to be replaced by its parent relation in order to reach the desired granularity:

- The parent relation *has affected feature* is defined as "a relation that holds between an instance of a genetic variation and a genomic feature (typically a gene class) that is affected in its sequence or expression". This indicates that an edge of relation *has affected feature* connects a node of concept variant with a node of concept gene since the dataset does not include any other concept that can be considered as a genomic feature. All children of this relation have been collected (Fig. 3.14). One of the children is the relation included in the original data as well being *is allele of*, while the other relations are not found between the concept entities.

The included relation labeled as *is allele of* has the definition of "A relation linking an instance of a variable feature (aka an allele) to a genomic location/locus it occupies which is typically a gene locus". Based on this definition, relation *is allele of* should exclusively link a variant to a gene taking into account all semantic groups that are available in the dataset.

Using *has affected feature* instead of *is allele of* is due to a divergence in the meaning of the connection between a variant and gene. For the relation *has affected feature*, the variant is either an allele of the gene or influences due to its altered sequence the expression of another gene. This could be the case with for example gene regulatory pathways in which one gene produces a protein that regulates the expression of another gene.

Following the current information granularity, the relation *is allele of* between a variant and gene indicates that this variant is a version of that gene, indicating both entities to be associated with the same functionalities, orthology relations and gene interactions excluding the consequences of the mutations that the variant brings. When a variant is linked with a gene by the relation *has affected feature*, it can be the case that the variant is associated with this gene in another way, having something to do with interaction between different genes as this variant does not necessarily contain an altered sequence of that connected gene. Thus, the relation *is allele of* cannot be replaced by its parent relation *has affected feature*.

In the built *domain model*, a gene and variant is also connected by the relation that implies that a variant is an allele of a gene. However, the relation *has affected feature* had not been included. Given the comparison between the definitions of the two relations *has affected feature* and *is allele of*, both relations are included in the *final model*: We cannot generalize *is allele of* to *has affected feature* associations because there is a risk of losing information as the specification between these relations is informative given the granularity.

- Two genes can be orthologous to each other in a one-to-one and in a one-to-many manner. The former means that both genes of the orthologous pair do not have more genes in the other species to which they are orthologous. The latter implies that one gene of the pair has more orthologous genes in the other species. This would mean that the gene has been duplicated in the other species after the speciation divergence.

  Based on the goals set during defining of the scope, it is only needed to convey the information that orthologous genes most likely share the same functionalities. The distinction between one-to-one, one-to-many and many-to-many orthologous gene pairs should not carry much more information that would benefit reaching the goal of yielding drug repurposing predictions and generating clear explanations. Thus, the child relation *in 1 to 1 orthology relationship with* is discarded and replaced by its parent relation *in orthology relationship with.* This change complies to the *domain model*, in which also a single relation exists between a gene and its ortholog.

- The relation *contributes to condition* is “a relationship between an entity such as a genotype or genetic variation and a condition represented as a phenotype or disease where the entity has some contributing role that influences the condition”. The other relation *causes condition* implies “a relationship between an entity such as a genotype or genetic variation and a condition being a phenotype or disease where the entity has some causal role for the condition”. From these definitions some statements can be made.

  Firstly, both relations and their subrelations connect some subject with an object that is exclusively belonging to either the disease or phenotype concept. The object can be a gene, gene variant or a genotype.

  The distinction between the relations *contributes to condition* and *causes condition* lays in the correlative and causative nature, respectively. It can be said that these relations are needed to be distinguishable in the *final conceptual model* as well, since the relations carry the information of a genetic entity having a certain level of effect on the condition. Now, it needs to be investigated whether their subrelations carry information that are sufficiently distinguishable from each other taking into account the defined scope.

  – The relation *is causal germline mutation partially giving rise to* “relates a gene to condition, such that a mutation in this gene partially contributes to the presentation of this condition.” Attempting to decrease the density of relations between concepts, we criticize that there is insufficient distinguishment between the parent relation *contributes to condition* and this subrelation. Specifically allowing the relation *contributes to condition* to have as subject an entity of concept gene, genotype or variant, the merging of the relation into its parent relation is enabled.

  – The relation *is causal germline mutation in* is a child of relation *causes condition* and “relates a gene to condition, such that a mutation in this gene is sufficient to produce the condition and that can be passed on the offspring”. The relation indicates that a mutation in this gene is causal to the condition which is already indicated by its parent relation *causes condition*. The information that distinguishes the two relations is that the mutation of the gene can be passed on to offspring. This is exceeding the granularity concluded from the goal modelling step and thus, this subrelation can be replaced by its parent.

  - For the relations *pathogenic for condition* and *likely pathogenic for condition* annotated definitions could not be found. Based on their labels, it is very closely resembling the information held by *causes condition* giving no reason to keep these subrelations being explicitly different from their parent relation. However, it can be observed that the subrelations differ in level of certainty of the causal effect on a condition. This distinction will be included in the *final conceptual model* by introducing a relation labeled as *likely causes condition*.

  In this way, the certainty level is made explicit while also removing relations that hold similar information used in the drug repurposing predictions. In the *domain model*, the difference between causality and correlation of a genetic entity as well as the certainty level have not been included. Due to its presence in the dataset and consideration of their importance, it has been realized that these relations are useful to include to the *final conceptual model*.

The analysis of shared ancestors between relations did not yield information for all relations in the original dataset. Some other relations that connect the same pairs of semantic groups are looked into as well:

- The relation labeled as *is marker for* with URI *RO:0002607* has the following definition: “C is marker for d if and only if the presence or occurrence of d is correlated with the presence or occurrence of c, and the observation of c is used to infer the presence or occurrence of d”. This relation is very closely resembling the relation *contributes to condition* that also indicates a correlation of an entity with a condition. Due to this resemblance, it has been chosen to replace *is marker for* with *contributes to condition*.

- The relations *involved in* (*RO:0002331*), *enables* (*RO:0002327*) and *is part of* (*BFO:0000050*) have been found to connect the same semantic groups. “An entity c involved in p holds if and only if c enables some process p’, and p’ is part of p”. “An entity c enables p if and only if c is capable of p and c acts to execute p”. The relation *is part of* is a “core relation that holds between a part and its whole”. Based on the original dataset, all three relations connect an entity to an entity that is exclusively of semantic group physiological process. Ideally, these relations are distinguished from each other on the basis of linking processes to other processes forming complex processes built of atomic ones as was done in the *domain model*. For example, the relation *involved in* could be removed and replaced by the entity connecting to the subprocess instead. However, the *data-based model* does not include any relations linking physiological process entities together. This indicates a lack of information to distinguish between the three relations in a conceptually clearer way. Based on their definitions, *enables* has a higher level of participation of the entity in the process than *involved in.* The relation *involved in* has a stronger association in participation than *is part of.*

- Relation *contributes to* (*RO:0002326*) is considered to hold the same information as relation *contributes to condition* given the observation that both relations have the same pair of semantic groups, being gene and condition which is either a disease or phenotype.
- Relation *has phenotype* (*RO:0002200*) is "a relationship that holds between a biological entity and a phenotype. The subject of this relationship can be a genomic entity such as a gene if modifications of the gene causes the phenotype." This suggests a strong similarity with the relation *causes condition* that can serve as the substitute relation.

Inconsistencies

By comparing the two drafted models and taking a thorough look at the definitions of all relations found between the concepts, some inconsistencies were found in the *data-based model* looking at all existing triples (Table C.1):

- The relation *contributes to condition* links genotypes and genes to a genotype entity. As previously determined, the relation can only have a phenotype of disease as the object.
- The relation *has phenotype* links orthologs to genotype entities while the definition of the relation only allows phenotypes as objects.
- The relation *has affected feature* has restrictions that are violated looking at the triples found in the original dataset. Based on its definitions mentioned before, the subject can only be a genetic variation while a triplet for this relation is present with as subject a genotype entity.
- The relation *has genotype* (*GENO:0000222*) is "a relationship that holds between a biological entity and some level of genetic variation present in its genome. The biological entity can be an organism, a group of organisms that shares a common genotype or organism-derived entities such as cell lines or biospecimens." In the dataset this relation has as subject exclusively a genotype and the object can be an ortholog, genotype or disease. Given the definition of the relation, these triples are incorrect. Interestingly, this relation indicates a new concept that has been overlooked before, as there are no concepts included yet that represent any form of the mentioned biological entity examples.

- The relation *in (1 to 1) orthology relationship with* is allowed to only link gene entities together when complying to the meaning of being in an orthology relationship. It is also clear from the *domain model* that the orthology relationship is expected to only be relevant to genes. The original dataset shows inconsistencies for these relations, because its triples also contain genotype entities as subject and object.
- The relation *is allele of* should have as subject a variant entity and as object an entity of the semantic group gene. However, the original dataset has a triplet including this relation and the possibility of the subject to be a disease entity.
- The relation *source* (*dc:source*) has something to do with "a related resource from which the described class/term's annotations are derived". This definition indicates the presence of entities that represent a publication. However, the concept publication is not and should not be included in both draft models as it will not provide information that is helpful for the drug repurposing predictions. The triplet that contains this relation has as subject physiological process entities and as object gene entities indicating some inconsistencies in the dataset.

The inconsistencies found in the *data-based model* are excluded from the *final conceptual model* by solving the problem that has caused this. In order to solve the problem, it should be investigated how these inconsistencies have occurred. The data fetching step used in previous research [57] has been revised to find out why these inconsistencies are present in the dataset. The outcome of this revision is described in one of the next sections.

Undefined Relations

There are triples with undefined relations. It is possible to speculate what information these relations may carry by referring to the *domain model*. The conceptual inconsistencies in the dataset will complicate making correct inferences, increasing the risk of forming incorrect assumptions about the meaning of these relations. So, this step will be performed after the problem of conceptual inconsistencies in the dataset has been solved.

3.2.4 Recreation of Data Fetching

The first step of the restructuring process has already been performed in order to verify the cause of the conceptual inconsistencies found in the original dataset. The data fetching steps from the Monarch Initiative Data platform, Therapeutic Target Database and DrugCentral have been recreated resulting in the connection graph shown in Figure 3.15. All triples found in this *new* dataset are shown in Table C.2. During revising the data fetching steps of the original data instances, the concepts of each entity has been derived from the prefix of its identifier. This method might result in unreliable categorization of entities as the meaning of the identifier prefix can change throughout time. This is why this step has been replaced by directly fetching the category assigned to the entity by the data platform itself.

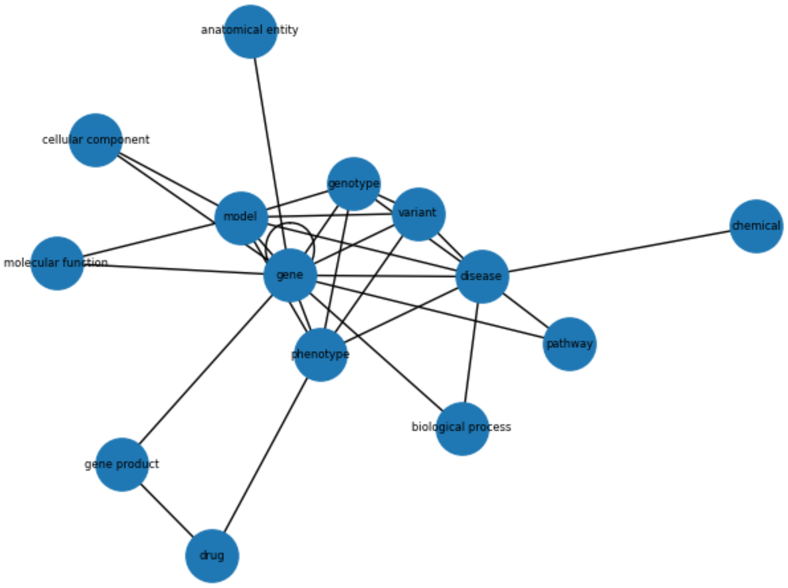


Figure 3.15: A graph in which each edge represents one or more relations that exist between the two connected concepts present in the dataset acquired by the newly built fetcher.

The concepts gene, variant and genotype have a set of relations in common. This enables categorization of these concepts that will be indicated as biological entity. This set of shared relations has to do with the correlative or causative associations with a phenotype or disease.

Due to acquiring the categories of the data entities from the source instead of deducing it from the identifier prefixes, some new concepts appeared in the new dataset that were not present in the original dataset. Thus, we need to figure out how these new concepts fit onto the *data-based model.* This can be done by comparing the triples of the original dataset with the ones of the new dataset, as we do know that the entities of the new concepts were categorized as other concepts in the original dataset. Also, it is investigated how the new concepts will be incorporated into the *final conceptual model.*

- The concepts cellular component, molecular function and pathway were not included in the original dataset. The entities categorized as cellular component are grouped into the concept physiological process in the original dataset. This is the case because both categories are found in triples with relation *is part of.* The entities of concept molecular function are connected to other entities by the relation *enables* which indicates that the original dataset has included these instances in the physiological process semantic group as well. Given the identified triples in the new dataset, the relation *involved in* exclusively connects a gene or disease to a pathway or a biological process. These findings oblige us to retract the previous statements about the three relationships carrying a different level of participation to a biological process as it now shows that the usage of relations merely differs based on the concept the object entity has. This changes the perspective as concepts have appeared that were not considered in either the data-based and *domain model*. As the concept physiological process from the original dataset has been split into four different semantic groups, it has been realized that these entities carry less information than initially thought. Given the connection graph in Figure 3.12, the concept physiological process seemed to associate with several other semantically grouped entities. Considering the triples of the new dataset, it can be seen that the concepts molecular function and cellular component are only associated with genes. The goals that need to be reached for yielding drug repurposing predictions (Fig. 3.6 and 3.7) indicated that connections between these concepts and phenotypes or diseases are needed to enable inferences. These goals cannot be reached with the current dataset. Despite this, the concepts are still included in the *final conceptual model* as they can contribute to finding genes similar to other genes. The concepts pathway and biological process are both able to be associated to genes as well as diseases. It has been chosen to merge these concepts together and consider this as the concept biological process since their meanings are quite similar: A pathway can be seen as a process in itself.

- The concept model did not appear in the previous data model. Also, this concept has not been thought of during the design of the *domain model*. Looking at the relations the model entities appear with, it might be possible to infer what this semantic group entails. It has been noticed before that the relation *has genotype* indicated a concept group that was not present in the original data model. In fact, this concept group would have to do with an organism or cell line as biological entity, being the required group of entities as subject of this relation. However, there are also indications that the concept model can be of the same nature as other concepts when looking at the constraints of some relations. The concept model appears as subject of relations that are heavily associated with the concept gene, such as *enables*, *in 1 to 1 orthology relationship with*, *interacts with* and *is part of*. Also, the concept also appears as a subject for the relation *has phenotype*. Due to this, the concept model is included in the *final conceptual model* as a new anti-rigid class including all biological entity concepts as members of this class. During the restructuring step, the entities originally assigned to model might be reorganized and distributed over all the biological entity concepts based on the set of relations the entities are associated with. The anti-rigidity comes from the fact that a model entity is only considered a model when it has a role in modeling at least one disease. When an entity that is originally of concept model is associated with a genotype or other variation via relation *has genotype*, the entity will be an instance of a new concept. This new concept needs to represent a collection of cells or an organism of a certain species in order to accord with the definition of *has genotype*. In the *final conceptual model*, this new concept is called organism/cell line.

As it was noticed before, there are some undefined relations between some concepts. The meaning of a few of these relations can be hypothesized while others cannot:

- The undefined relation between the pairs genotype and gene, variant and model, model and gene found in the new dataset (Table C.2) cannot be inferred given the *domain model*.
- The undefined relation between variant and genotype entities (Table C.2) does exist in the *domain model* as it depicts which variants are contained by which genotype. The relation will be expressed as *is variant in*.
- The meaning of the relation between chemical and disease might be hypothesized as well. The concepts chemical and drug are similar to each other, because a drug is a chemical component. Thus, the relation *is substance that treats* will be used and the concept chemical is replaced by drug.

Resulting Model

The complete overview of the *final conceptual model* can be seen in Figure A.3. For better understandability, the view is split into several parts.

The following statements have been expressed in the model in the partial view provided in Figure 3.16:

- A *gene* can interact with other *genes* or be colocalized with other *genes*.
- A *gene* is found in one specific *species*.
- A *gene* can be *ortholog* to another *gene* of another *species*.
- A *variant* is a variant version of one *gene*.
- A *variant* can affect the expression or sequence of a *gene*. The latter option should ideally be indicated by connecting a *variant* entity with a *gene* entity using the relation *is allele of*. However, it cannot be ruled out that a *variant* being the variant version of a *gene* is expressed by the relation *has affected feature* given the data instances.

- A *genotype* consists of one or multiple *variants* of different *genes*.
- An organism or cell line is of a *species* and is known to have a *genotype* or its genome to contain a certain *variant*.
- *Genes*, *variants*, *genotypes* and organisms/cell lines are considered to be *biological entities*.

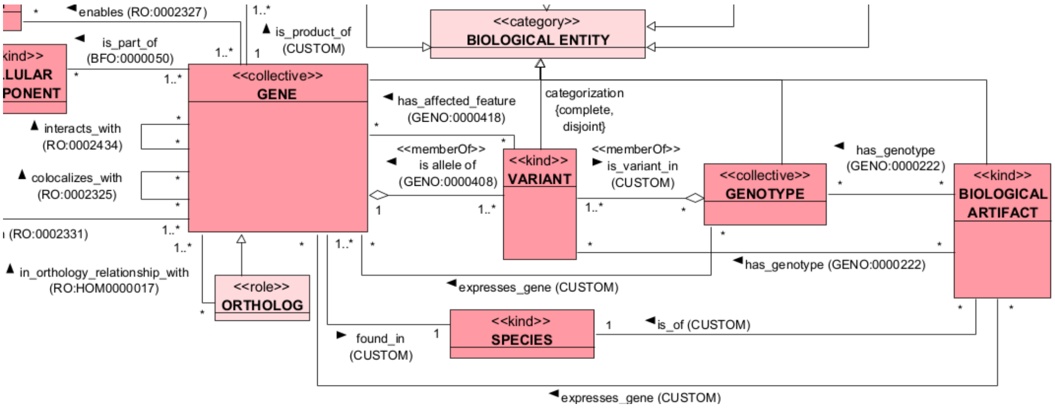


Figure 3.16: A partial view of the *final conceptual model* expressed with OntoUML. This part of the model is related to the biological entity concepts.

The next statements are expressed in another part of the model shown in Figure 3.17:

- A *gene* can be part of some *cellular components*.
- A *gene* can enable some *molecular functions*.
- A *gene* or *disease* can be involved in some *biological processes*.

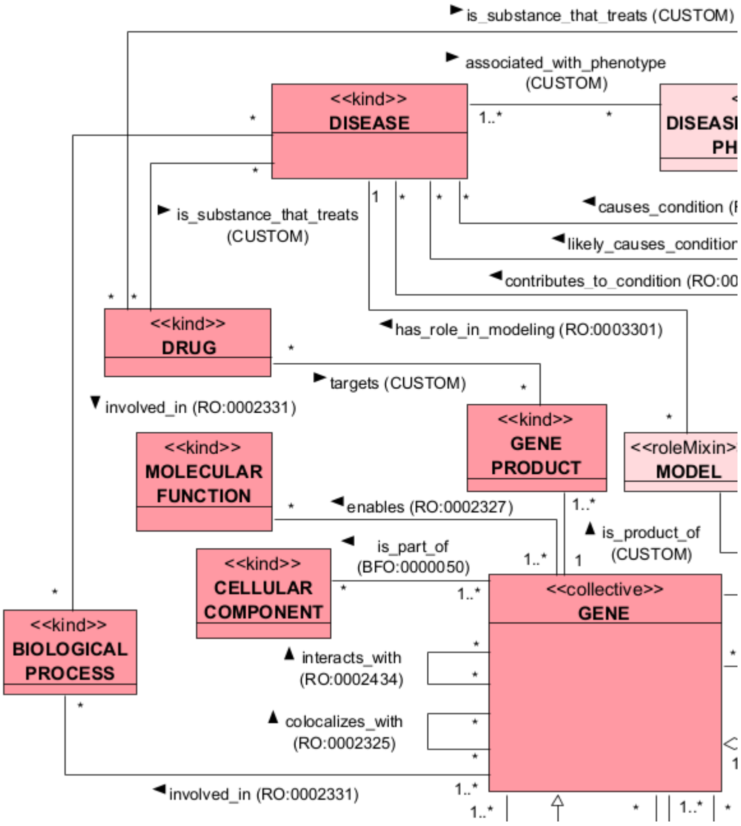


Figure 3.17: A partial view of the *final conceptual model* expressed with OntoUML. This part of the model consists of the concepts related to biological processes.

In Figure 3.18, the following statements are expressed:

- A *phenotype* can be associated with one or multiple *diseases*.
- A *drug* can be a substance that treats a *disease* or *phenotype*.
- A *drug* can target one or multiple *gene products* which are products of *genes*.
- A *biological entity* is a model when it has a role in modeling at least one *disease*.
- A *biological entity* can be correlated with or has a (likely) causal role in the appearance of a *disease* or *phenotype*.

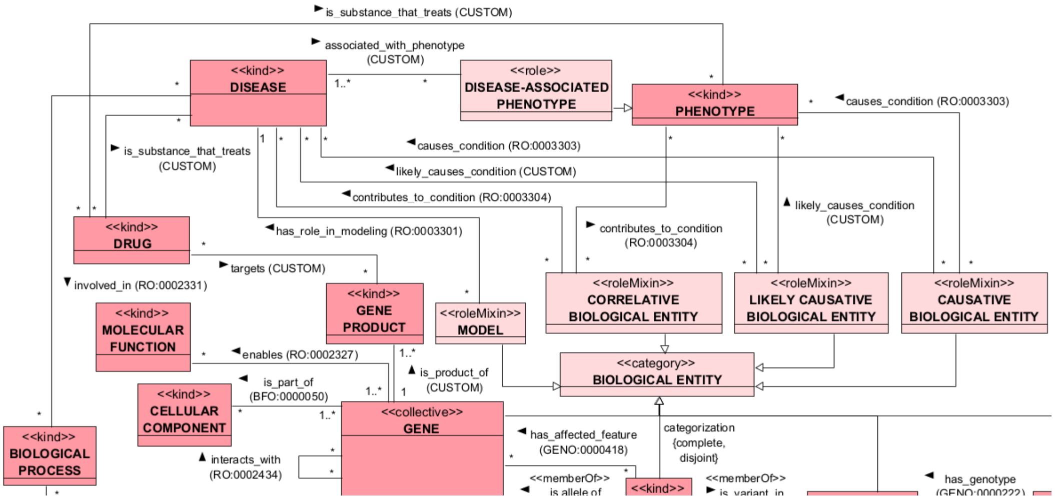

is_substance_that_treats (CUSTOM)
associated_with_phenotype (CUSTOM)
<<kind>> DISEASE
<<role>> DISEASE-ASSOCIATED PHENOTYPE
<<kind>> PHENOTYPE
causes_condition (RO:0003303)
is_substance_that_treats (CUSTOM)
causes_condition (RO:0003303)
likely_causes_condition (CUSTOM)
contributes_to_condition (RO:0003304)
has_role_in_modeling (RO:0003301)
contributes_to_condition (RO:0003304)
likely_causes_condition (CUSTOM)
<<kind>> DRUG
targets (CUSTOM)
involved_in (RO:0002331)
<<kind>> GENE PRODUCT
<<roleMixin>> MODEL
<<roleMixin>> CORRELATIVE BIOLOGICAL ENTITY
<<roleMixin>> LIKELY CAUSATIVE BIOLOGICAL ENTITY
<<roleMixin>> CAUSATIVE BIOLOGICAL ENTITY
<<kind>> MOLECULAR FUNCTION
enables (RO:0002327)
is_product_of (CUSTOM)
<<category>> BIOLOGICAL ENTITY
<<kind>> CELLULAR COMPONENT
is_part_of (BFO:0000050)
<<collective>> GENE
<<kind>> BIOLOGICAL PROCESS
interacts_with (RO:0002434)
has_affected_feature (GENO:0000418)
categorization {complete, disjoint}
<<memberOf>> is allele of
<<kind>>
<<memberOf>> is variant in
has_genotype (GENO:0000222)
1..*
1
1..*
1..*
1

Figure 3.18: A partial view of the *final conceptual model* expressed with OntoUML. This part of the model is related to the concepts disease and drug.

3.3 Drug Repurposing Pipeline

Using a knowledge graph as input, a graph neural network can be trained. In order to acquire predictions and explanations for these outcomes, several steps have to be taken, as shown in Figure 3.19b.

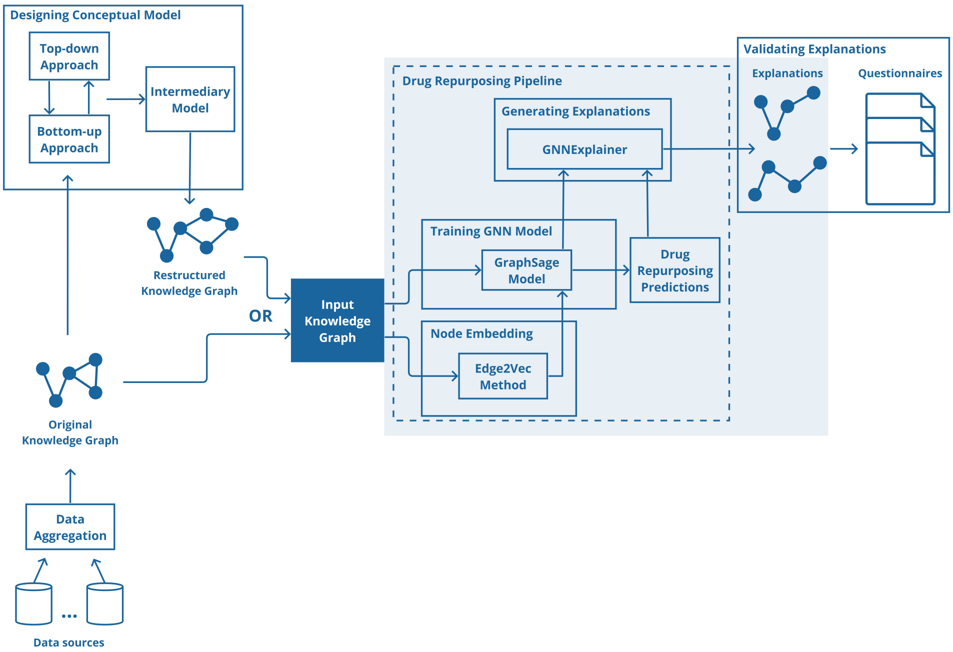
Before a graph neural network model is trained on the input graph, the characteristics of the nodes need to be obtained and represented. These features need to capture the heterogeneous neighbourhood of a node which can for example help express the semantic similarities or differences between other nodes in the graph. The use of the embedding methods provides a way to efficiently extract the characteristics of the nodes into a low dimension feature space despite having to deal with a graph that contains a large number of nodes and edges.

After this node embedding step, the GNN model can be trained such that new links can be predicted in the given input graph between a node that represents a phenotype or disease, and a drug node. Explanations are as follows generated for the most interesting drug repurposing predictions by applying the GNNExplainer method onto the trained GNN model.

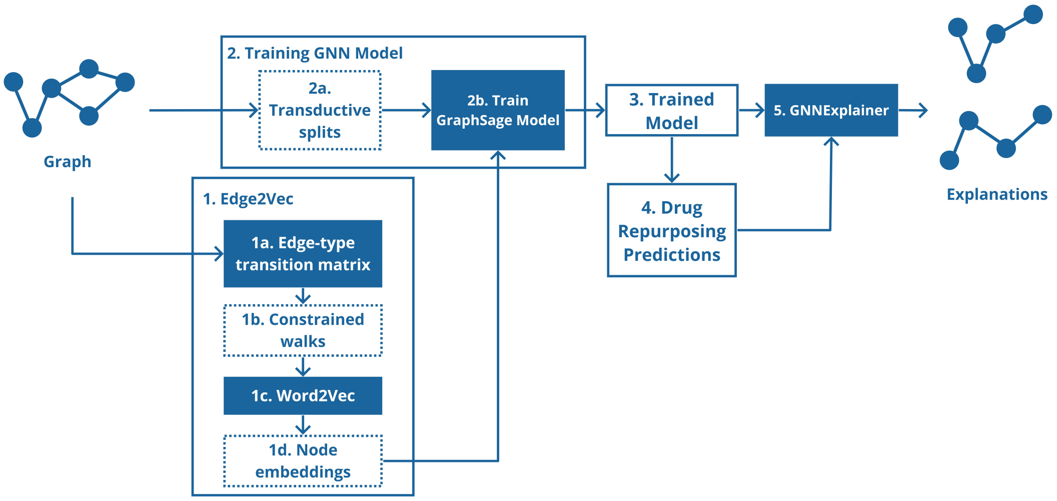
In the next sections, the mentioned steps and their used methods will be discussed in more detail.

3.3.1 Node Embedding

The input graph for this prediction task is a knowledge graph which means that the graph holds various types of nodes and edges. These types majorly



(a) This is a copy of the summary of all steps of this project seen in Figure 3.1. The steps highlighted with the blue rectangle are the processes that will be discussed in this section. In Figure 3.19b the details are shown of these steps.



(b) A diagram of the steps that are taken to acquire predictions and their explanations. For the graph to be compatible as input for the graph neural network, embeddings of the nodes are acquired that represent the heterogeneous structure of their neighbourhoods in the graph. The training, validation and test sets are created using transductive splits. After the graph neural network is trained, predictions are yielded relevant to the drug repurposing problem. Utilizing the GNNExplainer method, explanations are generated for the acquired predictions.

Figure 3.19: The diagrams that clarify the details and the position of generating the drug repurposing predictions and explanations in the overall project pipeline.

convey the meaning of the data as different relations need to be expressed between different biological concepts. The semantic nature of this data is crucial in order to represent its complex knowledge domain and thus, using a node embedding method that retains the rich heterogeneity of the graph is a priority. That is why the edge2vec [21] method is used.

For the first step, this graph representation method yields an edge-type transition matrix that contains the weights of transitions between all different edge types during a random walk process (Fig. 3.19b, step 1a). In this way, edge types that occur in the graph less frequently while being highly informative can still appear often in random walks because they have high transition weights with other edge types. Thus, the topological structure as well as the edge semantics of the network are considered. For calculating the edge-type transition matrix, the first step is setting all transition weights to the same value such that transition probabilities between all edge types are equal. The transition weights are then optimized following the Expectation-Maximization (EM) principles. In the Maximization step (M-step), a collection of random walk paths is generated constrained by the current edge-type transition matrix. In the Expectation step (E-step), the edge-type transition matrix is updated and optimized using the newly obtained biased random walks as feedback. From these walks, the number of occurrences of each edge type per path can be extracted and used for calculating the pairwise correlation scores between all edge types. A high correlation between a pair of edge types translates to a high transition probability between these edge types. The correlation scores are normalized such that the values are restricted to the transition probability range from 0 to 1. Now, these normalized scores are used as the updated transition weights of the edge-type transition matrix. The Expectation-Maximization steps can be iterated until the transition matrix is optimized sufficiently (Fig. 3.19b, step 1b). The total number of iterations that is needed for optimization that yields the best performance depends on the application and thus, needs to be included in the parameter tuning process. There are other parameters that need to be optimized related to the creation of the edge-type transition matrix such as the number of walks per node and the length of these paths in the M-step.

The second step of the edge2vec method is the training of a skip-gram model known as word2vec (Fig. 3.19b, step 1c). This is done by generating a series of biased random walks following the transition probabilities given by the previously optimized edge-type transition matrix. Skip-gram models are neural network-based models generally used for generating word embeddings. These embeddings represent words such that the words that are found in similar contexts are close to each other in the feature space. The architecture of the skip-gram model consists of one hidden layer. The input layer receives a one-hot encoded vector representing the target word. This vector has the size equal to the total number of unique words in the corpus. The neural network needs to predict which words have a high probability of occurring in the context of the target word. During the training process, the hidden layer learns the embedding representation of the given input target words. Thus, after training, this hidden layer provides the feature vector of the input word. In the case of node embeddings, the corpus is replaced by the collection of biased random walks in the form of node sequences. By using these paths, the skip-gram model learns to predict the contexts of the given target nodes. Now, the hidden layer of the trained model gives the node embedding which captures features of the target node. By inputting each node of the graph into the trained skip-gram model, the hidden layer can be used to extract the embeddings for all nodes in the graph (Fig. 3.19b, step 1d).

This node embedding process requires tuning of a set of hyperparameters to optimize the performance of this step in the pipeline. This set includes parameters such as the number of walks per node, the length of these walks and the probability expressed in p of returning to the previous node. Also, parameter q can be tuned, representing the probability that controls whether the random walks comply with a breadth-first or depth-first search.

The source code of the edge2vec method can be accessed from `https://github.com/RoyZhengGao/edge2vec`. This algorithm has been modified to work with Python 3 in the XAI-DMD-DR project. The resulting updated method is also included in this project and can be accessed at `https://github.com/rosazwart/XAIFO-ThesisProject`.

3.3.2 Training Graph Neural Network Model

For training the GNN the dataset needs to be split into a training, validation and test set (Fig. 3.19b, step 2a). For a link prediction task, the goal is to predict whether edges between nodes in the graph exist or not. To capture both the presence and absence of edges, negative sampling needs to be included. This means that all splits contain edges labeled as 0 or 1. The former means that the edge does not exist in the given graph and the latter represents an existing edge. For each positive sample in a split, one negative sample is taken from the graph. In this way, both classes are represented equally well. The used method for splitting the data is the transductive split [43]. This means that for each split, the whole input graph can be observed and used when computing the output of the model. The actual splitting is done on the labels of the edges which is a process with multiple steps in order to prevent any data leakage from the test set into the training or validation set. Firstly, the edges in the graph are divided into two types of edges being message and supervision edges. Message edges are fed into the GNN. The supervision edges are only used for determining the loss of the predictions throughout the training process of the model. The graph including only the message edges serves as the input graph of the GNN. The supervision edges are split into a training, validation and test set.

For the pipeline that splits the graph dataset into a training, validation and test set with negative sampling, the library *DeepSNAP* (Version 0.2.1) [1] is used.

3.3.3 Graph Neural Network Model Architecture

The architecture of the GNN consists of several GraphSAGE convolutional layers [32], each followed by a batch normalization layer. Each GraphSAGE convolutional layer samples and aggregates messages from the neighbours of each node. In this way, a function is learned that generates an embedding for any node that has been given as input [32]. The batch normalization layers allow the use of higher learning rates as it stabilizes the training process by normalizing the activation vectors of the hidden layers [35].

The number of layers that decides on the size of the neighbourhood considered in the message passing step of the GNN model is one of the hyperparameters that can be tuned to optimize the performance. Also, optimization can be done by tuning the hyperparameters such as the aggregation function, learning rate, the number of epochs and the size of the hidden and output layers.

For building the GNN model that will obtain the link predictions after training, several Python packages have been used. The tensor library for deep learning *PyTorch* (Version 1.11.0) [3] is applied to building the GNN model itself. For the implementation of the GraphSAGE module, the library *PyTorch Geometric* (Version 2.0.4) is used. This package has been built upon *PyTorch* in order to facilitate the building and training of GNNs.

3.3.4 Obtaining Predictions

The input of the GNN model are the node embeddings that have captured the heterogeneity due to using the edge2vec method. The GNN model outputs new embeddings for each node in a list. To acquire the prediction of an edge existing or not, the dot product[2] is calculated between the newly generated embeddings of the head and tail nodes. Next, the sigmoid function[3] is applied to this value, resulting in the confidence of the edge existing ranging between 0 and 1. The closer to 1, the higher the confidence of the prediction that the nodes are connected to each other in the graph (Fig. 3.19b, step 4). These confidence values can be compared to the labels of the supervision edges. Since the task

[2] The dot product receives two vectors with equal length n being $\mathbf{a} = [a_1, ..., a_n]$ and $\mathbf{b} = [b_1, ..., b_n]$. The dot product is defined as $\mathbf{a} \cdot \mathbf{b} = \sum_{n}^{i=1} a_i b_i = a_1 b_1 + ... + a_n b_n$. In this context, vectors $\mathbf{a}$ and $\mathbf{b}$ are the node embeddings where n is equal to the embedding size.

[3] The sigmoid function is commonly used in machine learning for transforming values to a probabilistic range. This logistic function has an S-shaped curve mapping any real-valued number to a value ranging between 0 and 1.

involves binary classification, the Binary Cross Entropy loss function [4] is used during the training process (Fig. 3.19b, step 2b).

3.3.5 Hyperparameter Optimization

Numerous hyperparameters from multiple subprocesses in the pipeline need to be tuned in order to optimize the performance of the GNN model on the knowledge graphs. The framework *RayTune* provides parallel processing of the hyperparameter tuning [44]. Random Search is applied to the hyperparameter optimization process. This means that for each search trial, a random combination of parameter values is used.

The version of *RayTune* that is used in this project is 2.3.1 and can be accessed at `https://docs.ray.io/en/releases-2.3.1/tune/index.html`.

3.3.6 Generating Explanations

For generating explanations given the trained graph-based deep learning model, the method GNNExplainer [67] is used (Fig. 3.19b, step 5). In this project, the GNNExplainer outputs a subgraph of the input knowledge graph that explains a given link prediction. This explaining subgraph is constructed by eliminating the edges that do not affect the outcome of the trained model. This is decided by looking at the difference between the confidence value predicted by the GNN using the full graph as input and the subgraph. If this difference is small, a subgraph has been found that contains all edges that have the most influence on the prediction result for the given link.

In practice, the GNNExplainer algorithm uses a binary mask over the edges to keep track of their importance. By masking an edge with a zero, it is not contained by the current subgraph. The optimization problem that the GNNExplainer method solves, is to obtain a mask representing a subgraph that maximizes the Mutual Information (MI) [67]. Here, MI quantifies the statistical dependence between the output of the GNN model and the inclusion or exclusion of edges in the input graph.

$$\max_{G_S} MI(Y, (G_S, X_S))) = H(Y) - H(Y|G = G_S, X = X_S) \tag{3.1}$$

In the formulation of the optimization problem given in the equation above (Eq. 3.1 [67]), G_S is the subgraph of the complete input graph of the GNN model Φ represented as G_C. For calculating MI, the entropy is considered of the initial prediction $H(Y)$ and the entropy of the prediction using the subgraph G_S. GNNExplainer can also consider importances of node features by including a subset of the node features to the optimization problem. However, in the drug repurposing input knowledge graph, there are no node features to consider in the explanation.

The entropy term $H(Y)$ in Equation 3.1 is fixed as the GNN model Φ is trained and outputs the same predictions given the same input. Thus, maximizing the mutual information can be simplified as it is equivalent to the minimization of the conditional entropy $H(Y|G = G_S, X = X_S)$. This implies that the found explanation for a prediction $\hat{y}$ is a subgraph and optionally a subset of node features that minimize the uncertainty associated with predicting this output $\hat{y}$ by Φ. In other words, the GNNExplainer finds the subgraph G_S and subset of node features X_S that maximize the probability of the GNN model Φ predicting $\hat{y}$.

The GNNExplainer method accepts parameters that set the size of the explanation subgraph. This parameter K_M will include the edges that give the K_M highest mutual information given the prediction. In this project the size has been kept equal to that of the XAI-DMD-DR project. To replicate the explanation generating step from this previous project as much as possible, the number of epochs used by the GNNExplainer and the learning rate to optimize on the mutual information are also kept the same.

The modified version of the GNNExplainer from the XAI-DMD-DR project is used as it added a function that enabled the generation of explanations for link prediction tasks. Also, the explanations that result from the GNNExplainer algorithm, are only accepted as obtained explanation when the subgraph is 'complete'. A 'complete' explanation is defined as a subgraph in which a path exists between the nodes that are connected by the predicted link that is explained.

The original version of the method comes from version 2.0.4 of *PyTorch Geometric* and can be accessed from `https://pytorch-geometric.readthedocs.io/en/2.0.4/_modules/torch_geometric/nn/models/gnn_explainer.html#GNNExplainer`.

3.4 Comparing Explanations

To answer our research question, the generated explanations need to be compared by measuring their explainability. This is a measurement that is best done by collecting subjective assessments from the intended user group of the DMD drug repurposing XAI system.

We acquired subjective evaluations from researchers that fall under the profile of a potential user of the drug repurposer by designing a questionnaire for evaluating the explanations. The requirement for participating in the survey is that the candidate needs to have a background in related to DMD or drug repurposing research. This ensures that answers are obtained from persons that can represent the intended user group and are grounded by relevant expert knowledge. To assess the level of explainability of a given explanation, a set of statements is developed to evaluate the different aspects of explainability being clarity, parsimony, completeness, and soundness in the perspective of the user:

- For assessing clarity, it has been asked whether the participant believes that the explanation is unambiguous regarding the use of concepts and relations.
- For assessing parsimony, it has been asked whether the participant believes that the explanation is presented in a way that is not too complex.
- For assessing completeness, it has been asked whether the participant believes that the explanation provides sufficient information to explain a new drug candidate.
- For assessing soundness, it has been asked whether the participant believes that the information paths shown in the explanations are useful for finding potential drug candidates.

The candidates need to assess these statements using Likert scaled scores [45] ranging from one to five where a score of one stands for strongly disagreeing with the given statement and five means that the person strongly agrees.

Considering the time constraint of the project and the specific candidate requirements, we anticipated on a limited number of participants being available to complete the questionnaires. Due to this, two questionnaires are created for both the model trained on the original knowledge graph and the model trained on the restructured knowledge graph. Each questionnaire consists of a single randomly chosen explanation from the set of all generated explanations.

The results of the distributed questionnaires will be shown and discussed in Section 4.4. Due to time constraints, we aim to collect more participants in future research in order to reach statistical significance for comparing the explanations.

3.5 Predictive Performance Metrics

The performance of the GNN model is measured by using the AUC-ROC score [53] which is a suitable performance metric when both the sensitivity and specificity of the model are important. A high sensitivity means that the model performs well in finding a large proportion of existing edges (true positives) while minimizing predicting positive edges being non-existent (false negatives). A high specificity shows that the model can find a large proportion of non-existing edges (true negatives) while it avoids classifying non-existing edges to be positive (false positives).

Some other performance metrics have been applied on the predictions of the trained model. The ROC curve is drawn showing the classification performance on the test set. The True Positive Rate (TPR, Eq. 3.2) and the False Positive

Rate (FPR, Eq. 3.3) are plotted against each other for different classification thresholds.

$$TPR = \frac{TP}{TP + FN} \tag{3.2}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.3}$$

The lower the classification threshold, the less edges are predicted as positive which leads to a lower TPR and FPR. A curve that approaches the upper left corner of the graph indicates a classifier with high accuracy as the sensitivity of the predictions approaches the maximum score of 1 and a maximal specificity [53]. The latter is the case because a FPR score of 0 equals a specificity of 1. The ROC curve has a close relation with the aforementioned AUC-ROC score, as this score is the area under the ROC curve. The closer the area under the curve is to 1, the more the curve fills the upper left corner of the graph.

Another measure is the F1-score in order to quantify the classification performance. In the F1-score the precision and recall are included with equal importance. The precision represents the proportion of correctly classified edges. A high recall indicates that a large proportion of edges labeled as a class, has been correctly predicted to belong to this class. By maximizing both the precision and recall, the model can identify all edges that belong to the current class while it does not assign this class to edges of the other class.

In Section 4.2.1 the Figures 4.7, 4.7 and 4.9 show the AUC-ROC score measurements used for assessing the performance of the GNN models. In Figure 4.10 the classification performance is expressed in F1-scores.

Chapter 4

Results

Results are presented that help assess the impact of the added methods to the drug repurposing pipeline of the XAI-DMD-DR project. The results are shown from the different steps of the whole method. First, the results of the graph analysis are presented to show the effect of the restructuring step on the structure of the knowledge graph. Next, the predictive performance and the predictions of the GNN models trained on both knowledge graph variants are shown enabling comparison on these aspects between the input variations. Lastly, objective and subjective measurements are used to analyse and compare the sets of explanations generated by the GNNExplainer.

In this section the term *original* is used to refer to the variation of the drug repurposing pipeline that attempts to replicate that of the XAI-DMD-DR project [57]. The term *restructured* is used for referring to the drug repurposing pipeline variation created during this project using the restructured knowledge graph as input.

4.1 Graph Analysis

The original and restructured knowledge graphs were compared using various measurements that assess the structure of the networks.

4.1.1 Node and Edge Composition

The occurrences of each node type in the graphs have been calculated and can be found in Tables 4.1 and 4.2. The restructuring of the graph results in the splitting of the concept physiological process represented by *PHYS* from the original network (Table 4.1) into several different concepts being *biological process*, *cellular component* and *molecular function* (Table 4.2). The original graph has a total of eight concepts and is increased in the restructured graph to 12 concepts. A total of 26 taxon nodes are included in the restructured graph showing that genes and biological artifacts originate from numerous different species. This information has been excluded from the original graph as it can only indicate whether a gene is found in either humans or non-human species depicted with *GENE* and *ORTH*, respectively.

The original knowledge graph consists of 72 triples shown in Table C.1 in the Appendix. The restructured knowledge graph has 37 triples and can be found in Table C.3.

For each node type, the average degree has been calculated including in- and out-degrees. While the node types *cellular component* and *molecular function* introduced in the restructured graph have the lowest presence rate (Table 4.2), these nodes appear to have very high average degrees. This indicates that although there are very few nodes belonging to these semantic classes, they are highly connected to other nodes. The highest average degree of 140.45 belongs to the concept *disease*. In the previous graph the node type *DISO* only has an average degree of 3.07. This can be explained by the restructuring step in which a distinction has been set between diseases and phenotypes. The nodes that represent diseases or phenotypes, are considered to be part of the same concept *DISO* in the original knowledge graph. In this concept class the disease nodes form the minority of the class, causing the average degree to differ between the node type *DISO* in the original graph and *disease* in the restructured graph. One of the seeds used to fetch data from the Monarch Initiative belongs to the *disease* node type and is the outlier with a degree of 1319 that causes the high average degree. The group of nodes that represents genes (*GENE* and *gene*) shows an increase in average degree after the graph has been restructured. This can be considered as a result of the alignment to the *final conceptual model* since new edges have been created between *gene* nodes and nodes that belong to the class *gene product* or *taxon*.

Table 4.1: Number of nodes per type, the percentage considering the total number of nodes and the average and median degree per node type in the original graph.

Node Type	**Count**	**Percentage**	**Average Degree**	**Median Degree**
DISO	5146	51.29%	3.07	1.0
ORTH	2880	28.70%	41.26	32.0
VARI	1125	11.21%	1.98	2.0
GENO	409	4.08%	12.17	2.0
DRUG	202	2.01%	1.23	1.0
GENE	202	2.01%	99.37	79.5
PHYS	50	0.50%	52.96	22.5
ANAT	20	0.20%	49.05	27.0

While the collection of concepts has been expanded, the restructuring process has led to a reduction of the number of edge types. To be specific, the original

Table 4.2: Number of nodes per type and the percentage considering the total number of nodes and the average and median degree per node type in the restructured graph.

Node Type	**Count**	**Percentage**	**Average Degree**	**Median Degree**
phenotype	5311	51.69%	2.86	1.0
gene	3163	30.78%	45.73	35.0
variant	1277	12.43%	2.30	2.0
drug	291	2.83%	1.45	1.0
biological artifact	71	0.69%	2.35	2.0
gene product	38	0.37%	10.39	5.0
genotype	36	0.35%	15.94	11.5
taxon	26	0.25%	124.38	154.5
biological process	24	0.23%	16.88	8.5
cellular component	17	0.17%	78.94	50.0
disease	12	0.12%	130.42	18.5
molecular function	9	0.09%	96.89	38.0

graph consisted of 24 different edge types, while the restructured graph has 21 types. The edge types with the lowest occurrences in the original graph (Table 4.3) are not found in the restructured graph (Table 4.4). The most occurring edge type *in 1 to 1 orthology relationship with* has also been excluded from the new graph as this edge type has been replaced by the relation *in orthology relationship with* as explained in Section 3.2.3. This results in a large majority of the edges in the graph to be of type *in orthology relationship with*.

Figure 4.1 indicates the overall connections found between concepts in the original graph (Fig. 4.1a) and in the restructured graph (Fig. 4.1b). Again, it can be observed that there are more concepts found in the new graph. Another interesting finding is that the conceptual restructuring resulted in less self-loops. In the original graph there were relations that connected pairs of nodes belonging to the same concept class. As has been discussed before in Section 3.2, these self-loops were conceptually incorrect given all present relations and their definitions.

4.1.2 Global Statistics

The node degree distributions of the knowledge graph before and after restructuring are shown in Figure 4.2. There are no striking differences between the distributions, except for a slight increase in frequency of nodes that have a degree total in the higher end. From node degree distributions, the class of the network can be determined. Scale-free networks are networks that show a decrease that approximates a power law when looking at the frequency of nodes with higher degrees [41]. Formally, the degree distribution $P(x)$ with x

Table 4.3: Number of edges per type and the percentage considering the total number of edges in the original graph.

Edge Type	**Count**	**Percentage**
in 1 to 1 orthology relationship with	29817	35.95%
in orthology relationship with	24020	28.96%
has phenotype	14520	17.51%
interacts with	7813	9.42%
is part of	1342	1.62%
has affected feature	1106	1.33%
pathogenic for condition	986	1.19%
expressed in	981	1.18%
enables	872	1.05%
involved in	405	0.49%
targets	239	0.29%
likely pathogenic for condition	185	0.22%
contributes to condition	177	0.21%
has role in modeling	134	0.16%
is allele of	96	0.12%
colocalizes with	60	0.07%
is substance that treats	31	0.04%
source	29	0.03%
is causal germline mutation in	16	0.02%
has genotype	7	0.01%
contributes to	4	0.00%
causes condition	3	0.00%
is marker for	1	0.00%
is causal germline mutation partially giving rise to	1	0.00%

representing the degree value, would thus be considered as:

$$P(x) \propto x^{-\alpha}$$

This property shows the presence of 'hubs' in the network which are the few highest-degree nodes connecting the large number of low-degree nodes together. A lot of real-world networks such as networks representing interactions in various contexts are classified as scale-free networks. In order to assess whether the original and restructured knowledge graphs can be considered to be scale-free networks as well, their degree distributions are fitted onto a power law distribution. The functionalities of the Python library powerlaw enabled us to find the parameter α for the degree distributions, yielding $\alpha = 1.81$ for the original graph (Fig. 4.3a) and $\alpha = 1.69$ for the restructured graph (Fig. 4.3b). The typical range of parameter α is $2 < \alpha < 3$.

Table 4.4: Number of edges per type and the percentage considering the total number of edges in the restructured graph.

Edge Type	**Count**	**Percentage**
in orthology relationship with	53837	62.61%
causes condition	15361	17.86%
interacts with	7813	9.09%
found in	3163	3.68%
is part of	1342	1.56%
has affected feature	1106	1.29%
enables	872	1.01%
contributes to condition	786	0.91%
involved in	451	0.52%
targets	357	0.42%
has role in modeling	188	0.22%
likely causes condition	185	0.22%
associated with phenotype	103	0.12%
is allele of	96	0.11%
is of	71	0.08%
is substance that treats	66	0.08%
colocalizes with	60	0.07%
expresses gene	56	0.07%
is product of	38	0.04%
is variant in	34	0.04%
has genotype	7	0.01%

Table 4.5: Global statistics calculated for the original knowledge graph and the restructured knowledge graph. Functionalities provided by the Python library networkx [31] have been used for calculation of the graph feature values.

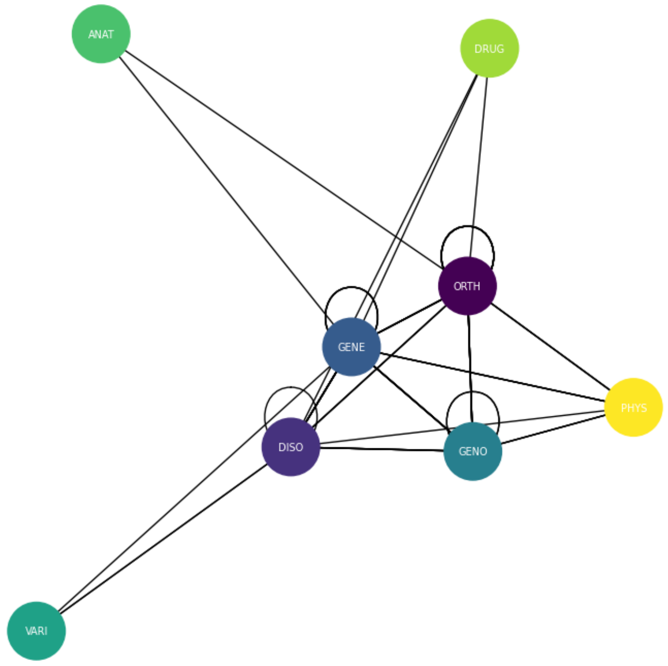
Property	**Original Graph**	**Restructured Graph**
Number of Nodes	10034	10275
Number of Directed Edges	82899	85878
Average Degree	16,524	16,716
Highest Degree	1660	1637
Diameter of Undirected Network	7	7
Average Shortest Path Length of Undirected Network	3,777	3,753
Average Clustering Coefficient	0,325	0,299

Other features of both graphs have been calculated using the functionalities from the Python library networkx [31] shown in Table 4.5. The restructured graph has an increased total of nodes and edges. The average degree of nodes in the restructured graph is slightly higher than the average degree in the original graph. The highest degree found in the network is highest in the original graph. Also, the diameter has been calculated which entails the shortest path length between the nodes that have the largest distance between each other in the graph network. This feature has been calculated on an undirected interpretation of the graph networks, because taking into account the directionality causes the graph to be disconnected. This shows that both directed graph networks are weakly connected which means that all nodes in the graph networks have an in- or out-degree of at least one. For the original and restructured graph, the diameter remains the same. The average shortest path length is very slightly lower for the restructured graph. Somewhat less clustering appears in the restructured graph in comparison to the original.

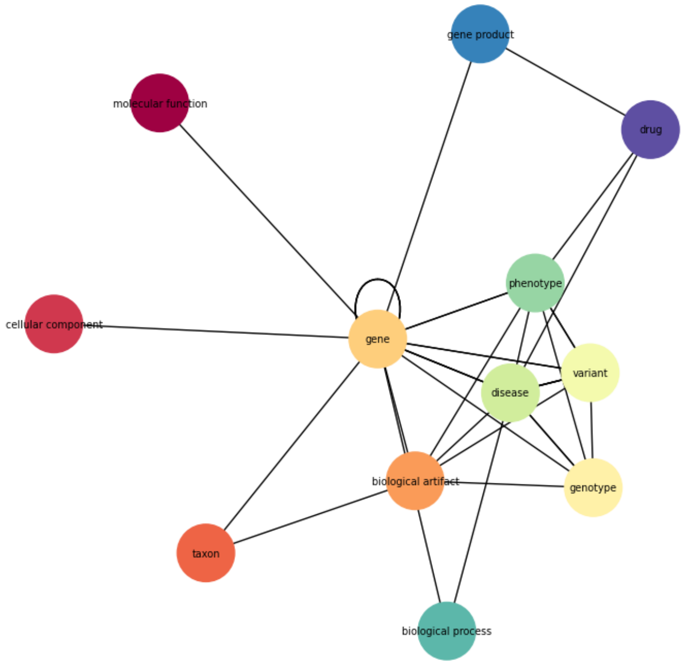
4.1.3 Graph Network Visualization

The open-source and free graph network visualization platform Gephi [9] has been used to explore the knowledge graphs visually and interactively. In order to get some idea about the structure of large networks, the layout algorithm Force Atlas 2 [36] has been run on both graph networks. Force Atlas 2 is a continuous force-directed network spatialization algorithm that is developed specifically for the application of the Gephi platform on scale-free networks containing 10 to 10,000 nodes. The network moves towards a layout during which nodes repulse each other and edges attract their nodes [36]. This layout enables the visual interpretation of the structures of the networks. Indeed, the visualization for the original graph (Fig. 4.4) shows a variety of clusters. A lot of small clusters have formed containing mostly ortholog nodes (ORTH). The disease nodes (DISO) can be found in a few very small clusters, but are mostly present in the middle of the network. The majority of the variant nodes (VARI) seems to be all connected to the same gene node and disease node. We can hypothesize that these two nodes represent the initial seed nodes for retrieving the data from Monarch Initiative explaining their high connectivity to the other nodes.

For the restructured graph (Fig. 4.5), similar clusters can be found. The nodes in the middle of the original graph belonging to the disease concept represent the same nodes of the cluster in the middle of the restructured graph. The phenotype instances have been considered to be part of the disease group in the original knowledge graph. Thus, this makes sense, as this cluster contains nodes that belong to the phenotype conceptual group. In a visual way, it becomes clear that the majority of the original disease node group represents phenotypes because nodes of the concept disease in the restructured graph are not apparent in the network visualization. The nodes belonging to the genotype conceptual group can be found in a cluster in the visualization of the restructured network, while the nodes of the similar semantic group (GENO) in the original graph are more dispersed. There is a prominent cluster containing variant nodes which is also the case in the original knowledge graph visualization. The most noticeable difference is the presence of the drug nodes in the outer area of the network layout. This has not happened during the formation of the network layout of the original knowledge graph which indicates some change in the structure of the graph. It can possibly be explained by the addition of the gene products in the graph, causing this structural shift. Lastly, the nodes that represent the seeds used for gathering the data from Monarch Initiative stand out much less than has been seen in the original graph network layout.

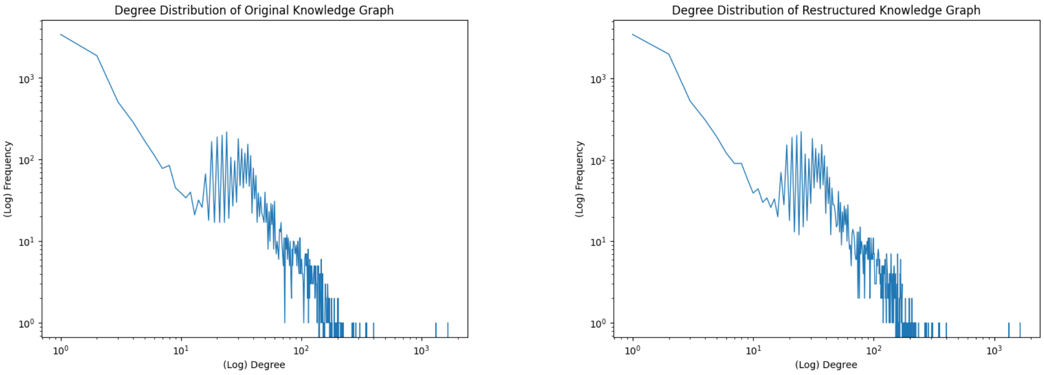


(a) Metagraph of the original graph.



(b) Metagraph of the restructured graph.

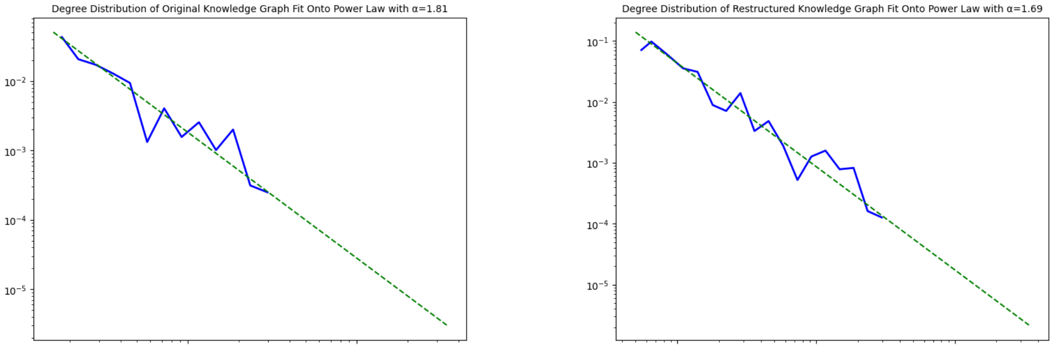
Figure 4.1: Metagraphs of the original and the restructured graphs showing the existing connections between concepts.



(a) Degree distribution of the original knowledge graph in logarithmic scale.

(b) Degree distribution of the restructured knowledge graph in logarithmic scale.

Figure 4.2: Degree distributions of the original and restructured knowledge graph showing the frequency of each degree in logarithmic scale.



(a) Degree distribution of original knowledge graph fit onto the power law distribution $P(x) \propto x^{-\alpha}$ with $\alpha = 1.81$.

(b) Degree distribution of restructured knowledge graph fit onto the power law distribution $P(x) \propto x^{-\alpha}$ with $\alpha = 1.69$.

Figure 4.3: Degree distributions of the original and restructured knowledge graph fit onto a power law distribution $P(x) \propto x^{-\alpha}$. The parameter α is found using the functionalities of the Python library powerlaw [7].

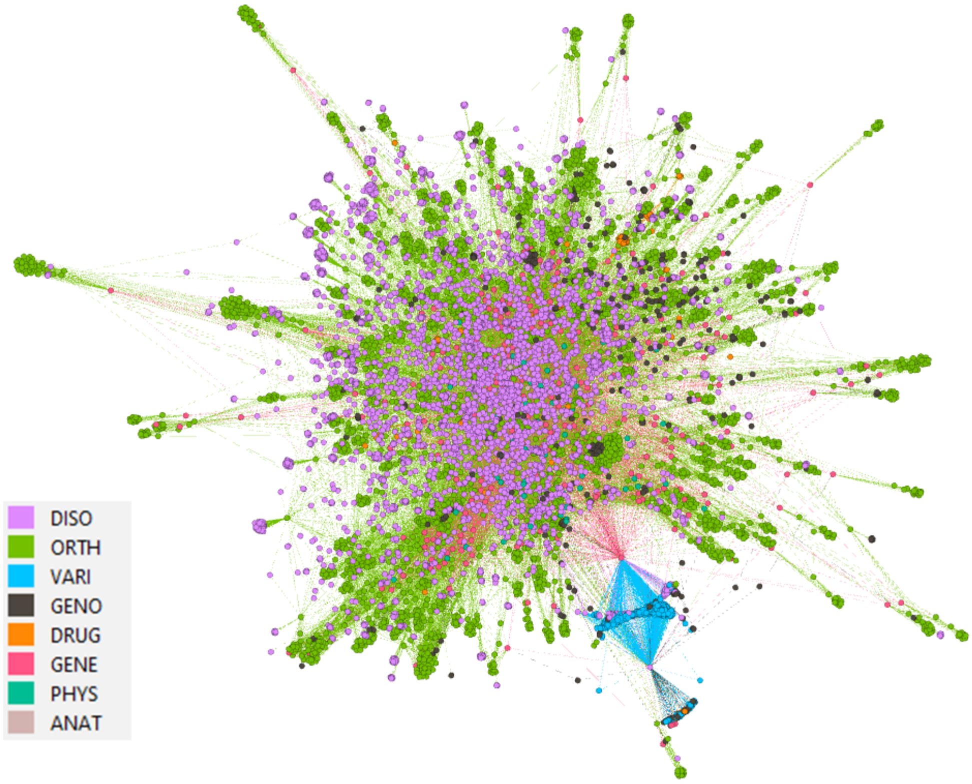


Figure 4.4: Graph network visualization of the original knowledge graph. The nodes have different colours based on their concept class indicated in the legend.

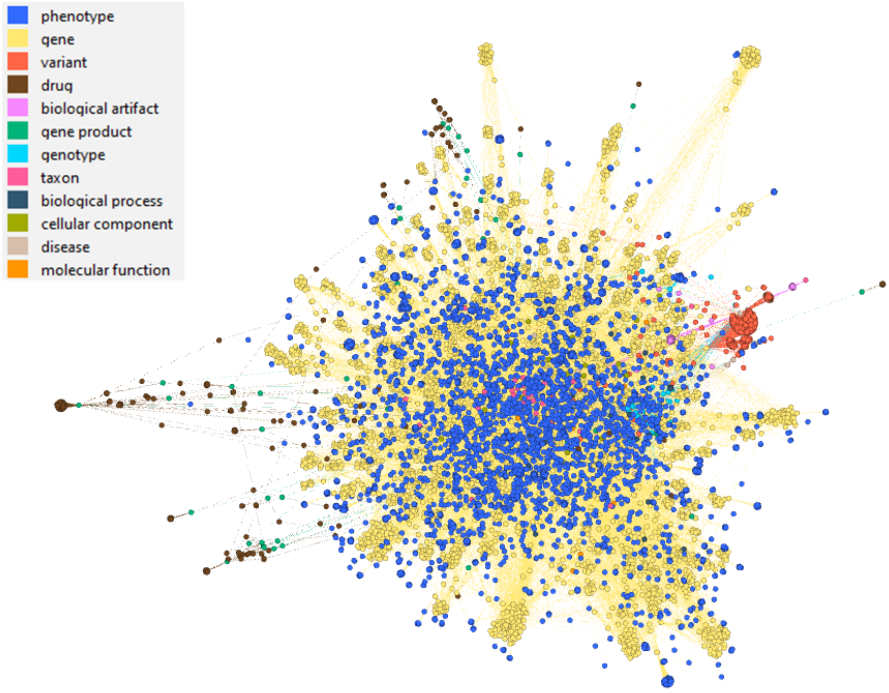


Figure 4.5: Graph network visualization of the restructured knowledge graph. The nodes have different colours based on their concept class indicated in the legend. There are no correlations between the colours of this visualization and the visualization in Figure 4.4.

4.2 Graph Neural Network Models

Although the same drug repurposing pipeline is used for both the original and restructured knowledge graphs as input, hyperparameter optimization has been applied for the GNN model that uses the restructured knowledge graph as training data. For the original knowledge graph, the hyperparameter settings are used that have been found in the XAI-DMD-DR project [57]. The resulting optimal hyperparameter settings are shown in Table 4.6.

Process	Hyperparameter	Search Space	Optimal Value for Dataset 1	Optimal Value for Dataset 2
Edge2Vec	Number of walks	2, 4, 6	2	6
	Walk length	3, 5, 7	7	7
	Embedding dimension	32, 64, 128	32	64
	p	0.50, 0.75, 1.00	0.70	0.75
	q	0.50, 0.75, 1.00	1	1
	Epochs	5, 10	10	5
GNN	Hidden dimension	64, 128, 256	256	256
	Output dimension	64, 128, 256	64	64
	Layers	2, 4, 6	2	2
	Aggregation function	mean, sum	mean	mean
	Dropout	0.0, 0.1, 0.2	0.2	0.2
	Learning rate	1e-4 - 1e-1	0.07000	0.01348
	Epochs	100, 150, 200	150	100

Table 4.6: The values for each hyperparameter found after hyperparameter optimization using *RayTune* [44]. The choices or range of each hyperparameter used during Random Search are included as well. Dataset 1 represents the original knowledge graph as input and dataset 2 the restructured knowledge graph as input of the GNN model.

To compare the performance, predictions and explanations resulting from the drug repurposing pipeline using the different knowledge graphs as input, the process is run independently ten times for each input dataset (Figure 4.6). This method takes into account the randomness that is present in the pipeline components such as in the node embedding algorithm and the graph neural network model. By averaging over multiple runs, comparison between the outcomes of the models trained on a different knowledge graph is facilitated, thereby creating a more robust analysis.

4.2.1 Performance

The performance of the drug repurposing predictors are measured and compared using different metrics looking at the training curve of the GNN model, F1-scores and ROC curve. In order to show the effect of the edge2vec method appliance prior to training the GNN model, additional drug repurposing predictor variations have been added to the performance comparison.

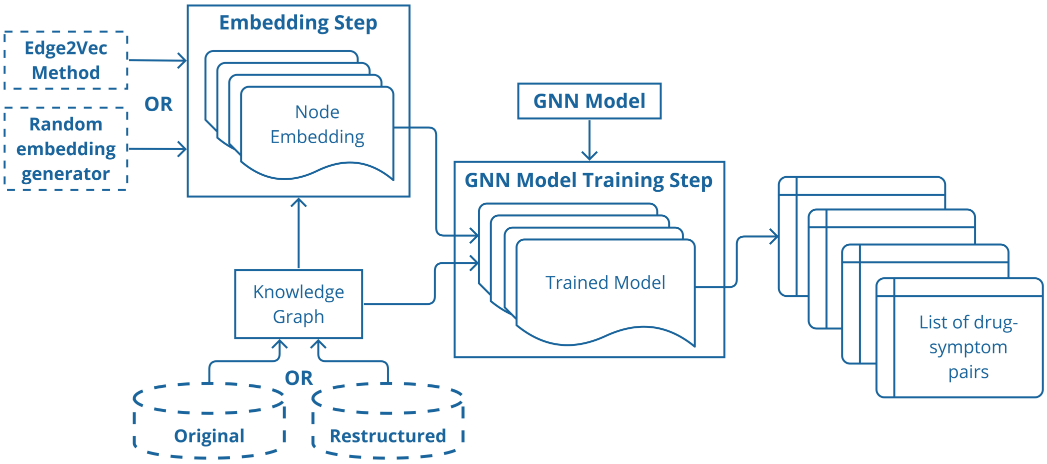
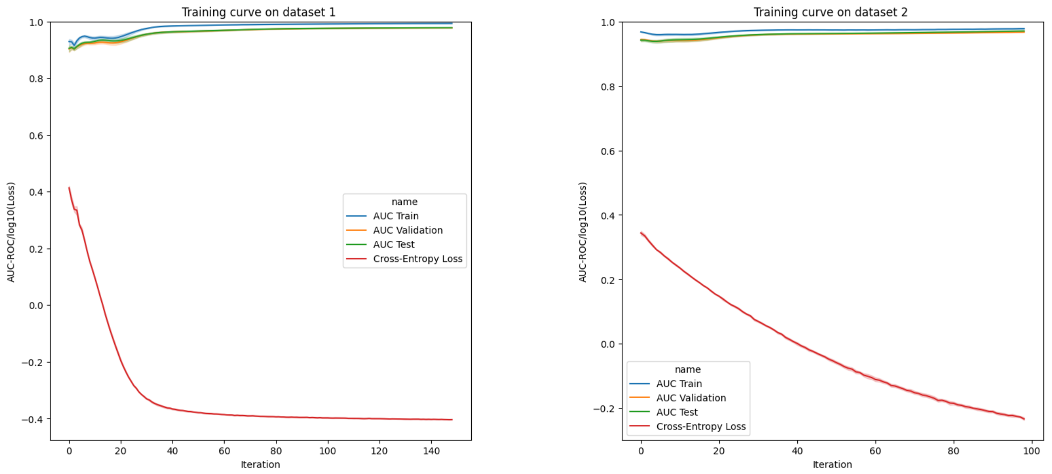


Figure 4.6: A diagram showing the process of acquiring the results from the drug repurposing pipeline performing multiple runs. The diagram boxes with the dashed lines show the elements of the process that can be swapped out for another embedding method or input dataset.

First, the performance scores are compared of the drug repurposing pipeline using the original and restructured knowledge graph as input data. For the second experiment, the pipeline has been modified by removing the edge2vec method from the process. Instead of performing this step, random embeddings are assigned to the nodes of the input graph. This option is also shown as a possibility in the diagram of Figure 4.6. In combination with replacing the edge2vec method with the random embedding generator, the restructured knowledge graph is used as input of the predictor. Based on the outcomes of the performance, the impact of adding an embedder method prior to training the GNN model on its predictive performance can be determined. In the last experiment, the default drug repurposing pipeline is kept. However, the restructured knowledge graph is changed by removing all relation types of the edges and replacing it by a single relation type. Now, the level of effect of different edge types present in the knowledge graph using the proposed drug repurposing pipeline can be identified.

Comparing Performance with Original and Restructured Knowledge Graph as Input

In Figure 4.7, the training curves are shown for the GNN model trained on the original (Fig. 4.7a) and on the restructured knowledge graph (Fig. 4.7b). For both the model with the original and restructured knowledge graph as input, the training process starts at an already remarkably high AUC-ROC score for both the training and test set. The accuracy of the test set gradually approaches the predictive performance on the training set. However, the improvement from the start to the end of the training process is minimal. Due to the small differences in AUC-ROC scores, close-ups are included for both model variations looking at Figures 4.8a and 4.8b. For the GNN model trained on the restructured graph, there is less improvement throughout the training epochs. The average AUC-ROC and F1 scores after training of all ten runs are compared between the two model variations in Figures 4.9 and 4.10.



(a) Training curve of the GNN model with the original knowledge graph as input.

(b) Training curve of the GNN model with the restructured knowledge graph as input.

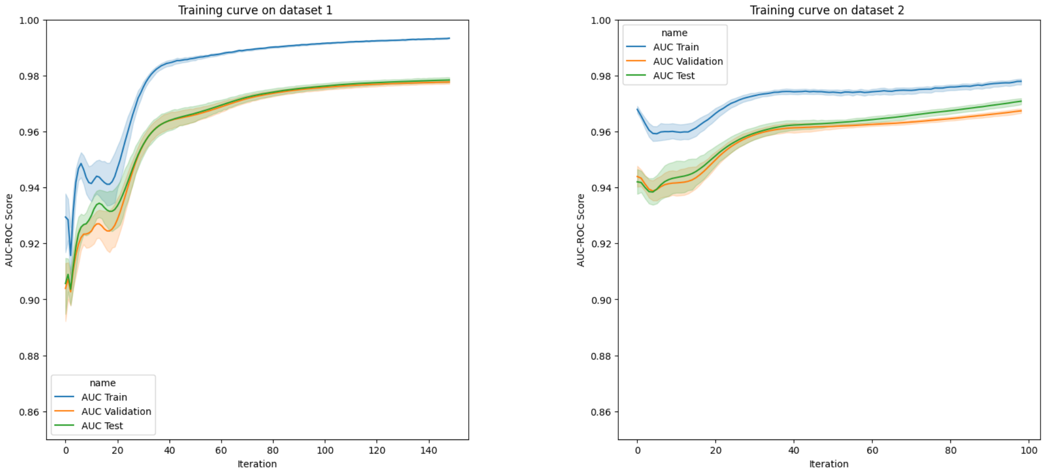
Figure 4.7: Training curves of the GNN models with edge2vec node embedding using the original and restructured knowledge graph as input. The performance throughout the epochs is measured in AUC-ROC scores and cross-entropy loss for the train, validation and test set. The average of the scores per epoch is represented by the solid lines and the deviation from the average measurements is shown by the colored areas.

Random Node Embeddings

In Figure 4.11 the training curve is shown of the GNN model that uses a graph with randomized node embeddings as input. In comparison to the performance of the default drug repurposing pipelines regardless of which graph is used as input, the predictions start off worse at the first epochs. The AUC-ROC score on the training set improves. However, the predictive accuracy on both validation and test sets decreases progressively throughout the training process indicating overfitting.

Knowledge Graph with Single Relation Type

Surprisingly, the performance of the GNN model with as input a knowledge graph containing just a single relation type for all edges is similar to that of the models trained on the restructured knowledge graph. The training curve of this drug repurposing predictor variation can be found in Figure 4.12. This



(a) Close-up of training curve of the GNN model with the original knowledge graph as input.

(b) Close-up of training curve of the GNN model with the restructured knowledge graph as input.

Figure 4.8: Close-ups of the training curves of the GNN models with edge2vec node embedding using the original and restructured knowledge graph as input. The performance throughout the epochs is shown in AUC-ROC scores for the train, validation and test set. The average of the scores taken from all ten runs per epoch is represented by the solid lines and the deviation from the average measurements is shown by the colored areas.

could indicate that the variation in edge types in the graph does not play any role in embedding the nodes.

4.2.2 Predictions

After training the models, the outcomes of the predictions are considered. For yielding the predictions of the drug repurposing models, the probability scores are calculated for the existence of the edge between nodes that represent drugs and symptoms. As it is needed to predict new drug candidates for symptoms that are related to DMD, a list of all drug and relevant symptoms found in the knowledge graphs needs to be acquired. For both the original and restructured knowledge graph, a total of 27 symptoms has been found that are associated with the disease DMD. The symptoms with their identifiers are shown Table 4.7. For the original knowledge graph, a total of 202 drugs are present. The restructured graph contains 291 entities of the drug class.

For each symptom that is related to the disease DMD, the three drugs with the highest probability of being a candidate for treating the symptom, have been collected. Thus, for each run of a drug repurposing predictor a list is created in which each relevant symptom is paired with three drug candidates.

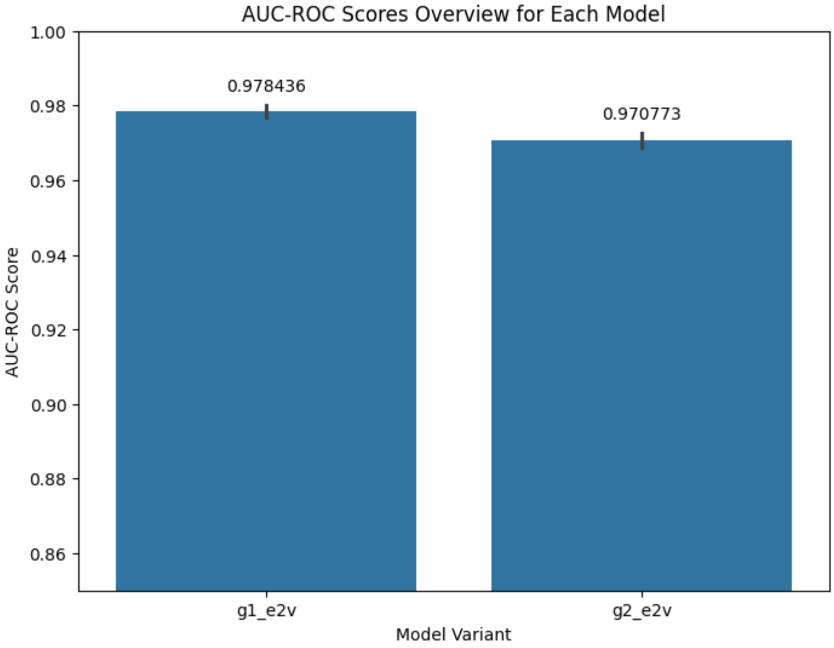


Figure 4.9: The average AUC-ROC score obtained by each model over ten runs. The standard deviation is shown by the vertical line at the top of the bar. g1_e2v: The average AUC-ROC score of the GNN model with edge2vec node embedding trained on the original knowledge graph after 150 epochs. g2_e2v: The average AUC-ROC score of the GNN model with edge2vec node embedding trained on the restructured knowledge graph after 100 epochs.

Comparing Predictions with Original and Restructured Knowledge Graph as Input

Each drug repurposing model variation has been run ten times which means there are ten lists of the top three drug candidates for each DMD-related symptom. To show the collective agreement on the prediction of new drug candidates, the drug-symptom pairs have been collected that are found in the obtained lists from all runs. For the drug repurposing model trained on the original knowledge graph, there are four common pairs (Table 4.8). The model using the restructured knowledge graph as input data has an overlap of six pairs considering all ten runs (Table 4.9).

The consistency of the prediction outcomes of the models needs to be verified to determine the reliability of their drug repurposing predictions. In this way, it can be found out whether the predictions are robust or simply random each time the model is run. This is achieved by pairwise comparison of runs for each model variation, determining the percentage of drug-symptom pairs that are identical in both predicted pair lists. Thus, the lower this percentage, the less overlap there is between the predictions of two runs indicating less consistent outcomes from the model. The overlap between each run is visualized using a

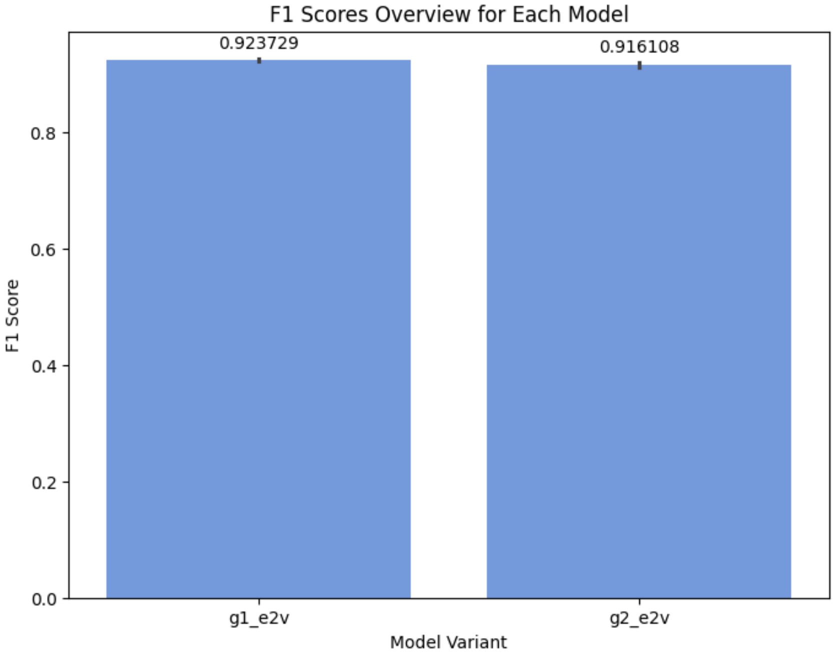


Figure 4.10: The average F1-score obtained by each model over ten runs. The standard deviation is shown by the vertical line at the top of the bar. g1_e2v: The average F1-score of the GNN model with edge2vec node embedding trained on the original knowledge graph after 150 epochs. g2_e2v: The average F1-score of the GNN model with edge2vec node embedding trained on the restructured knowledge graph after 100 epochs.

heatmap. The comparisons between the predictions of the runs of the model trained on the original knowledge graph are shown in Figure 4.13. For the restructured knowledge graph as input, the overlap percentages between all ten runs are visualized in Figure 4.14. Looking at the calculated mean and median values of the overlap of predicted drug-symptom pairs, the drug-repurposing model outcomes are more consistent when using the restructured knowledge graph as input.

The feasibility of a new drug candidate for treating a DMD-related symptom could be assessed by conducting literature research, searching for relevant case reports and collecting opinions from pharmaceutical and medical experts. To answer the research question posed by this project, we focus on the explanations in the context of explainability (discussed in Section 4.4). Therefore, validation of the drug-symptom pairs has not been done as it was outside the scope of the project.

Random Node Embeddings

The GNN model trained using input with randomly generated node embeddings performed poorly on the test set in contrast to the GNN models that

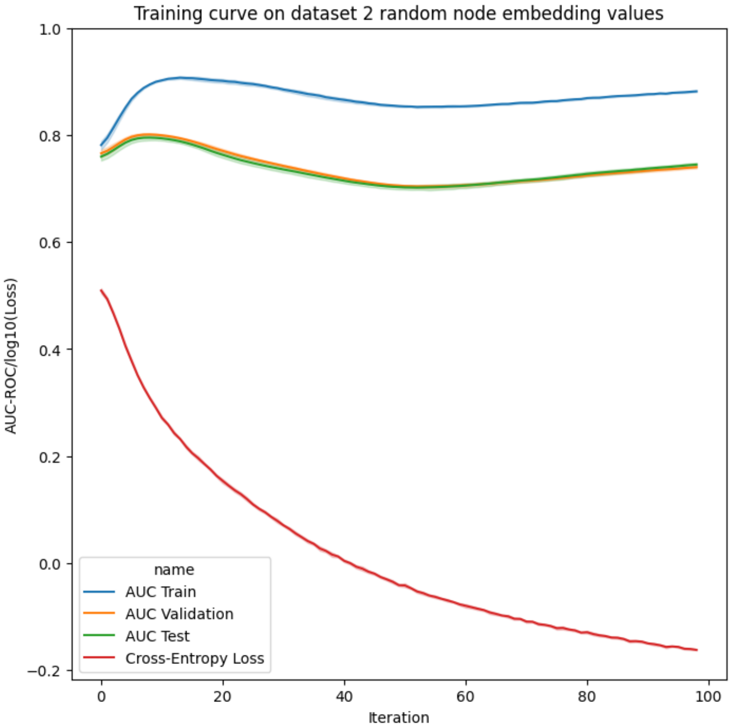


Figure 4.11: Training curves of the GNN model with randomized node embeddings using the restructured knowledge graph as input. The performance throughout the epochs is measured in AUC-ROC scores and cross-entropy loss for the train, validation and test set. The average of the scores per epoch is represented by the solid lines and the deviation from the average measurements is shown by the colored areas.

implemented the edge2vec method. The inferior performance carries over to the pairwise overlap of predicted drug-symptom pairs of all runs shown in Figure 4.15.

4.3 Explanations

After training the drug repurposing predictors, the trained models can be used as input of the GNNExplainer. For the GNNExplainer to generate explanations, specific edges need to be given for which an explanation will be calculated. As there are multiple runs for each model variation, the GNNExplainer is applied to each of these trained model instances. Only the edges are chosen as input of the GNNExplainer that are predicted by a defined number of runs.

Initially, the GNNExplainer has been run on the edges that are predicted by all ten runs for each drug repurposing model variation listed in Tables 4.8 and 4.9 for the original and restructured knowledge graph as input, respectively. An insufficient number of explanations was generated in which a path

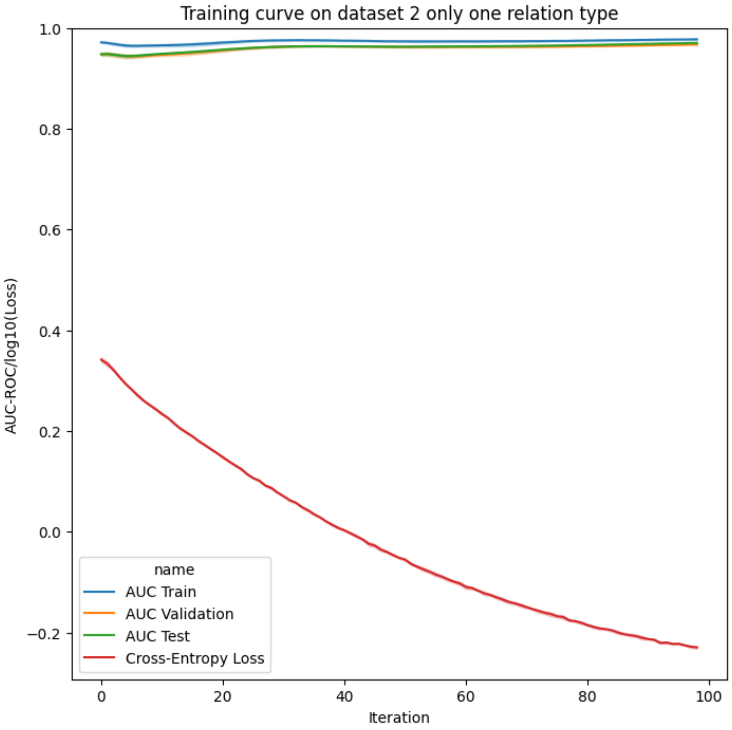


Figure 4.12: Training curves of the GNN model with edge2vec node embedding using the restructured knowledge graph in which all relation types are replaced by a single relation type as input. The performance throughout the epochs is measured in AUC-ROC scores and cross-entropy loss for the train, validation and test set. The average of the scores per epoch is represented by the solid lines and the deviation from the average measurements is shown by the colored areas.

exists between the drug candidate and the symptom it is predicted to treat. Due to this issue, the number of edges as input for the GNNExplainer needed to increase. This has been achieved by decreasing the threshold which means that the edges have been considered that were predicted by at least six out of the ten runs rather than all runs. This measure did not have to be taken for generating the explanations from the model trained on the restructured knowledge graph.

4.3.1 Generated from Original Knowledge Graph

A total of 10 explanations were generated from the predictor trained on the original knowledge graph. The acquired subgraphs explain the following predicted edges:

- Neratinib treats delayed speech and language development (Figures D.6, D.7, D.8, D.9 and D.10 in the Appendix)
- Neratinib treats global developmental delay (Figures D.1, D.2, D.3,

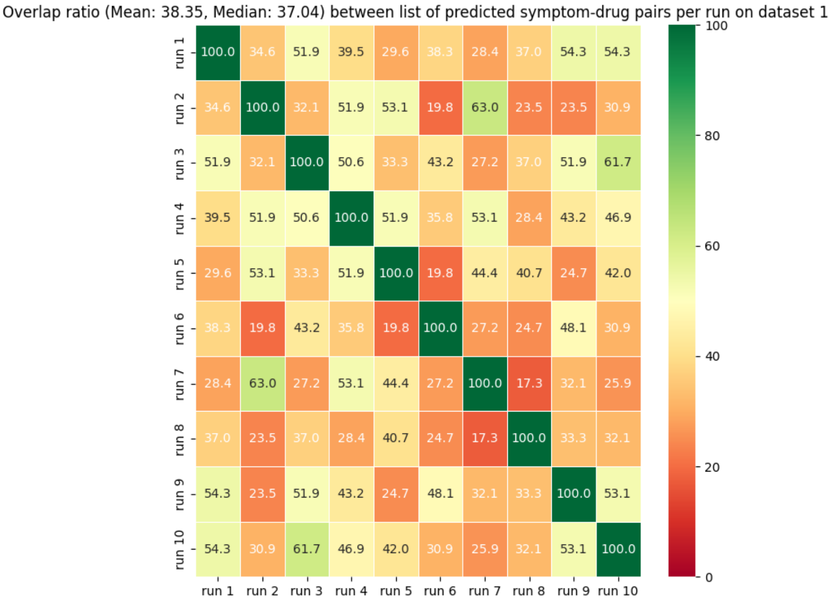


Figure 4.13: Heatmap that shows pairwise overlap of predicted drug-symptom pairs of all ten runs in percentages. The predictions are obtained by the drug repurposing model trained on the original knowledge graph. The mean of the overlap ratio is 38.35 and median is 37.04.

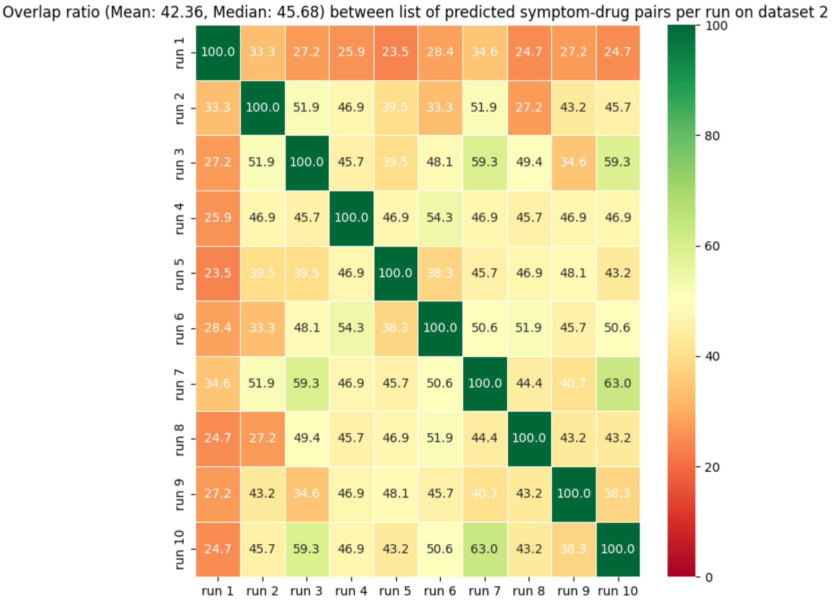
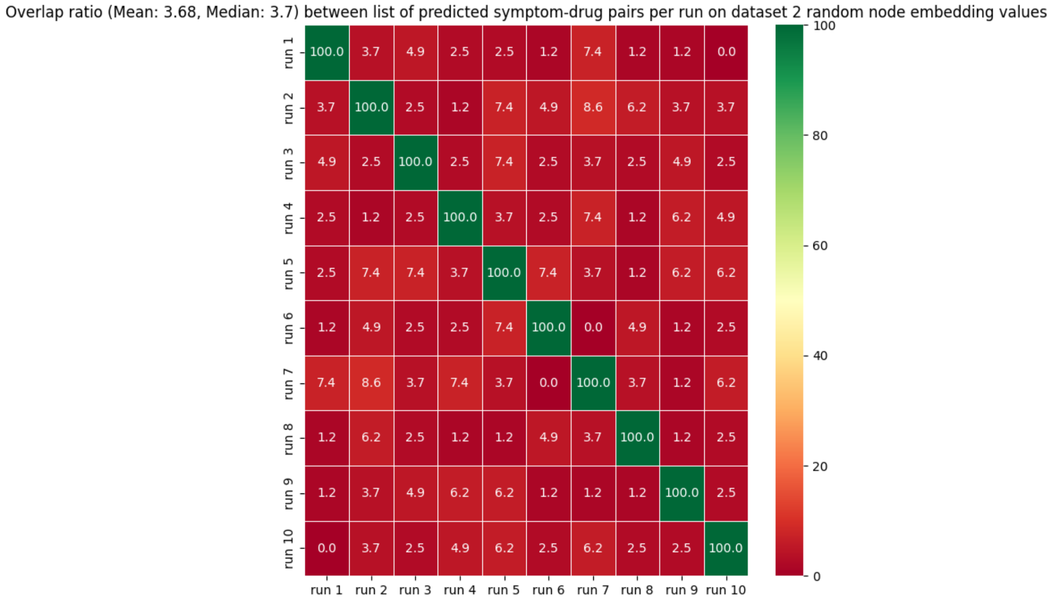


Figure 4.14: Heatmap that shows pairwise overlap of predicted drug-symptom pairs of all ten runs in percentages. The predictions are obtained by the drug repurposing model trained on the restructured knowledge graph. The mean of the overlap ratio is 42.36 and median is 45.68.



	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
run 1	100.0	3.7	4.9	2.5	2.5	1.2	7.4	1.2	1.2	0.0
run 2	3.7	100.0	2.5	1.2	7.4	4.9	8.6	6.2	3.7	3.7
run 3	4.9	2.5	100.0	2.5	7.4	2.5	3.7	2.5	4.9	2.5
run 4	2.5	1.2	2.5	100.0	3.7	2.5	7.4	1.2	6.2	4.9
run 5	2.5	7.4	7.4	3.7	100.0	7.4	3.7	1.2	6.2	6.2
run 6	1.2	4.9	2.5	2.5	7.4	100.0	0.0	4.9	1.2	2.5
run 7	7.4	8.6	3.7	7.4	3.7	0.0	100.0	3.7	1.2	6.2
run 8	1.2	6.2	2.5	1.2	1.2	4.9	3.7	100.0	1.2	2.5
run 9	1.2	3.7	4.9	6.2	6.2	1.2	1.2	1.2	100.0	2.5
run 10	0.0	3.7	2.5	4.9	6.2	2.5	6.2	2.5	2.5	100.0

Figure 4.15: Heatmap that shows pairwise overlap of predicted drug-symptom pairs of all ten runs in percentages. The predictions are obtained by the drug repurposing model trained on the restructured knowledge graph with randomized node embeddings instead of the embeddings generated from the edge2vec method. The mean of the overlap ratio is 3.68 and median is 3.7.

D.4 and D.5)

4.3.2 Generated from Restructured Knowledge Graph

A total of 17 explanations has been generated from the predictor trained on the restructured knowledge graph. The acquired subgraphs explain the following predicted edges:

- Levosimendan treats arrhythmia (Figures D.15, D.17, D.20, D.21, D.22, D.23, D.24, D.25, D.26 and D.27 in the Appendix)
- Levosimendan treats cardiomyopathy (Figures D.11, D.12, D.13, D.14, D.16, D.18 and D.19)

4.4 Comparing Explanations

The explanations from the different model variations are compared objectively and subjectively. For the latter, the average number of node and edge types in the explanation graphs is calculated. Also, the average number of triples found in the explanations are reported. Lastly, the average shortest path length between the drug candidate and the treated symptom is added to the measured features of the explanations. The values are found in Table 4.10. The explanations generated from the model trained on the restructured knowledge graph score higher for each of these features.

It has been observed that all explanations from the original knowledge graph connect the concept *drug* to the rest of the subgraph with the relation *targets*. Remarkably, the explanations from the restructured knowledge graphs connect the concept *gene* with the other nodes exclusively via the relation *is substance that treats*.

Looking at the explanations from both drug repurposing pipeline variations, it stands out that the explanations show information paths that do not seem to directly explain a potential link between a drug and symptom. This means that there are nodes and edges that are not found in the path between the node of the drug candidate and the node of the treated symptom. It can be considered that these parts of the explanations do not explain to a human user why a drug candidate could potentially treat a certain DMD-related symptom and can thus be considered irrelevant for the explanation.

The comparison that will majorly address the research question is the objective comparison that considers the explainability of the explanations. To perform this comparison, experts have been asked to fill in a questionnaire in which the explainability is rated using its aspects being clarity, parsimony, completeness and soundness. In order to retrieve well-thought ratings, the experts who were allowed to participate in the survey had background in research related to DMD or drug repurposing. Four questionnaires have been distributed over 10 experts with an average of 13 years of experience in research on DMD or drug repurposing. From the 10 experts, 8 researchers conduct research on the rare disease DMD.

Each questionnaire shows a randomly chosen explanation from the set of generated explanations from one of the two drug repurposing model variations. An overview of the questionnaires and their content is shown in Table 4.11. To facilitate interpreting the graphs of explanations, a short description has been added to describe the most relevant explaining paths in the graph. The participants can also click on some concepts such as genes to be directed to a page that provides information about the specific gene. The questionnaires are created using Google Forms and can be reviewed in Appendix E.1. For more detailed information regarding the responses, consult the tables included in Appendix E.2.

The ratings given to explanations for each input knowledge graph variation have been merged together in order to assess the differences in explainability between the explanations from the drug repurposing pipeline using the original input and the set of explanations from the model trained on the restructured knowledge graph. The mean ratings given for each aspect of explainability are given in Table 4.12.

To verify a statistically significant difference between the ratings for each model variation, the unpaired t-test has been applied which quantifies the significance of the difference in the mean value of two independent groups by taking into account the distribution of the ratings [65]. In this comparison, the null hypothesis is that the ratings for an explainability aspect do not differ between the sets of explanations that come from different drug repurposing models. Ideally, this null hypothesis should be rejected in order to accept the alternative hypothesis that says that the two sets of explanations result in different ratings for the explainability aspect. These hypotheses are considered for each explainability aspect while calculating the P value. Specifically, the two-tailed P value has been chosen to consider for determining the significant difference for the mean rating of one set being larger or smaller than the other. The acquired P values are shown in Table 4.12. Considering the most adopted cut-off point of 0.05 where $P < 0.05$ concludes significant evidence against the null hypothesis [14], the null hypothesis cannot be rejected for each of the explainability aspects. This means that for each explainability aspect, we cannot say that the mean rating for one set of explanations is significantly different from the mean of the ratings for the other set.

Although the differences in the average ratings for each aspect of explainability are not considered significant, we can cautiously conclude that the explanations generated from the drug repurposing pipeline using the original knowledge graph as input were rated higher on average for clarity, parsimony and soundness.

Given the standard deviations of the ratings, there is an overall larger dispersion in given ratings for the explanations generated using the restructured knowledge graph.

Participants were allowed to add comments to the explanation they had rated. These comments can be found in Table E.2. To summarize, for both sets of explanations, participants had difficulty interpreting the graph despite offering a short description of the relevant explaining paths in the given explanation and references to the information pages of the concepts present in the graph. One participant commented that the accompanying description actually made it more difficult to understand the given graph. Another participant mentioned that they expected links between certain concepts that were not present in the explanation. To be specific, the gene *DMD* was not linked to the disease DMD. Another participant mentioned the lack of context of interactions as it is important to know whether interactions lead to inhibition or activation. They also missed information about a drug's side effects, which might affect the outcome of drug repurposing predictions.

For an explanation resulting from the drug repurposing pipeline using the original knowledge graph as input, one participant commented on the labels used to categorize the concepts. They suggested using terms that are more commonly understood, as the term ‘DISO’, for example, has an unclear and ambiguous meaning.

Table 4.7: The list of 27 symptoms found in the original and restructured knowledge graph that are associated with the disease DMD.

Symptom ID	**Symptom**
HP:0011675	Arrhythmia
HP:0002515	Waddling gait
HP:0003236	Elevated serum creatine kinase
HP:0002093	Respiratory insufficiency
HP:0003707	Calf muscle pseudohypertrophy
HP:0003701	Proximal muscle weakness
HP:0003202	Skeletal muscle atrophy
HP:0003560	Muscular dystrophy
HP:0003391	Gowers sign
HP:0001635	Congestive heart failure
HP:0001328	Specific learning disability
HP:0003323	Progressive muscle weakness
HP:0001371	Flexion contracture
HP:0002650	Scoliosis
HP:0003115	Abnormal EKG
HP:0001263	Global developmental delay
HP:0008981	Calf muscle hypertrophy
HP:0001638	Cardiomyopathy
HP:0003307	Hyperlordosis
HP:0000750	Delayed speech and language development
HP:0001265	Hyporeflexia
HP:0001644	Dilated cardiomyopathy
HP:0001270	Motor delay
HP:0001290	Generalized hypotonia
HP:0100543	Cognitive impairment
HP:0002791	Hypoventilation

Table 4.8: The drug-symptom pairs of which the edge has been predicted with a top three probability scoring for all ten runs of the GNN model trained on the original knowledge graph.

Drug	**Symptom ID**	**Symptom**
levosimendan	HP:0011675	Arrhythmia
aprindine	HP:0003115	Abnormal EKG
aprindine	HP:0001635	Congestive heart failure
entrectinib	HP:0002650	Scoliosis

Table 4.9: The drug-symptom pairs of which the edge has been predicted with a top three probability scoring for all ten runs of the GNN model trained on the restructured knowledge graph.

Drug	**Symptom ID**	**Symptom**
levosimendan	HP:0001644	Dilated cardiomyopathy
levosimendan	HP:0001638	Cardiomyopathy
azathioprine	HP:0001290	Generalized hypotonia
levosimendan	HP:0003115	Abnormal EKG
levosimendan	HP:0003236	Elevated serum creatine kinase
levosimendan	HP:0011675	Arrhythmia

Table 4.10: Features calculated for the explanations generated from the model trained on the original and the restructured knowledge graph. These features give an idea about the conceptual enrichment of the explanations as well as the shortest information path length between the drug candidate and treated symptom node.

	Average number of node types in explanations	Average number of edge types in explanations	Average number of triples in explanations	Average shortest path length between drug candidate and treated symptom
Original Knowledge Graph	3.0	4.5	5.0	2.6
Restructured Knowledge Graph	4.6	5.5	8.0	3.3

Table 4.11: An overview of the explanations selected to be shown in each questionnaire.

Questionnaire	Input Knowledge Graph of Trained Model	Added Explanation Explains	Reference to Explanation
#1	Original	neratinib treats delayed speech and language development	Figure D.9
#2	Original	neratinib treats global developmental delay	Figure D.3
#4	Restructured	levosimendan treats arrhythmia	Figure D.25
#5	Restructured	levosimendan treats cardiomyopathy	Figure D.14

Table 4.12: The mean values of the ratings given to both sets of explanations and the standard deviation (SD). The first set of explanations generated from the drug repurposing model trained on the original knowledge graph. The second set contains the explanations generated from the predictor trained on the restructured knowledge graph. All ratings for each set of explanations are averaged over 5 responses. Also, an unpaired t test has been performed of which the two-tailed P values are shown.

	Explainability							
	Interpretability				Fidelity			
	Clarity		Parsimony		Completeness		Soundness	
	I believe that the explanation is unambiguous regarding the use of concepts and relations.		I believe that the explanation is presented in a way that is not too complex.		I believe that the explanation provides sufficient information to explain a new drug candidate.		I believe that the information paths shown in the explanation are useful for finding potential drug candidates.	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Explanations using Original Knowledge Graph	2.80	1.10	3.40	1.82	2.60	0.55	3.80	0.45
Explanations using Restructured Knowledge Graph	2.40	1.14	2.60	1.34	2.80	1.10	2.80	1.64
Two-tailed P Value	0.5871		0.4511		0.7245		0.2256	

Chapter 5

Discussion and Conclusion

5.1 Discussion

Changes were made to the method of gathering the information from the Monarch Initiative data platform for building the restructured knowledge graph. These changes could not be avoided as the information fetcher from the XAI-DMD-DR project omits too many classes into which the acquired biological entities are categorized making it impossible to restructure the knowledge graph to the extent of what we envisioned from the conceptual modelling approach. Ideally, the knowledge graph variations are built using the same fetching method to ensure that both knowledge graphs contain the exact same edges and nodes fetched from Monarch Initiative which has not been the case during this project. The results may also be influenced by the fact that the information for the original and restructured knowledge graphs was not collected on the same day. There is a risk that the Monarch Initiative data platform has undergone updates between the two dates on which the original and restructured knowledge graphs have been created leading to a difference in acquired nodes and edges. This could have been avoided by being more aware about this risk and fetching the information for both knowledge graphs as quickly as possible in succession.

When training the GNN model with both the original and the restructured knowledge graph as input, the predictions already have a surprisingly high accuracy in the first epochs. To assess whether this high initial performance is caused by the node embedding step prior to training the model, an experiment has been done during which the node embeddings from the edge2vec methods have been replaced by randomly generated node embeddings. The training curve on this variation of the GNN model shows a very different progression as the accuracy starts lower (Fig. 4.11). This might indicate that the node embedding process prior to training the GNN model has a positive effect on the training as it provides a good starting point for the node feature values.

The robustness of the GNN model trained using the randomly generated node embeddings as initial input is much lower compared to the consistency of the predictions found during independent runs of training the other GNN model variations. This is another indication that the node embedding step plays a crucial role when it comes to performance and consistency of the GNN model.

Since the preceding node embedding step implementing edge2vec seems to have a positive impact on the performance of the drug repurposing pipeline, the application of the edge2vec method has been investigated in more detail. This is done by finding out whether the GNN model has less use of the generated node embeddings as input when removing the edge heterogeneity of the knowledge graph. Interestingly, the GNN model performs equally well (Fig. 4.12) compared to the accuracy observed during the training process of the model using the restructured knowledge graph (Fig. 4.7b). The edge2vec method is known for its ability to capture the heterogeneity of edges in a given knowledge graph into the generated node embeddings. It could be said that a change would be noticed in performance of the GNN model when the node embedding includes or excludes any information about the heterogeneity of the input knowledge graph. However, this difference in performance has not been observed and thus, might indicate that the edge2vec method does not fully utilize the conceptual structure of a given knowledge graph.

We observed that the GNN model shows less improvement in accuracy during training on the restructured knowledge graph than on the original knowledge graph (Fig. 4.8). This might indicate that the node embeddings carry less useful information from the restructured knowledge graph compared to the node features generated from the original input, resulting in the GNN to learn less effectively. This is a plausible reason as the edge2vec method has already been suspected to not fully utilize the presence of the different relations in the restructured knowledge graph. Another reason would be that increasing the learning rate might help with improving the increase in accuracy improvement as a lower learning rate can cause a slower convergence. During this project we did perform hyperparameter tuning on the drug repurposing pipeline. However, it is never guaranteed that this Random Search method will find the most optimal settings as its results are probabilistic.

While the average ratings on the explainability aspects were on the lower side for the explanations from the restructured knowledge graph, other evidence has been found that shows a positive effect of having conceptually restructured the knowledge graph. We found that the restructured knowledge graph as input of the drug repurposing predictor leads to more consistent predictions over multiple independent runs. This improvement might be considered as a promising finding and a reason to investigate further on the effect of different conceptual structures of the input data on the predictions of a GNN. However, it could also be a side effect of a less welcoming reason. For example, a possible cause of this increased consistency would be that an additional node has been added between a *drug* that targets a certain *gene* being a node that represents an instance of the concept *gene product*. Currently, the knowledge graph does not include information about gene products interacting with other products. The inclusion of the *gene products* as concepts in the knowledge graph thus results in increasing the information distance while not adding more information in the form of additional connections to other data instances. This increase in information distance may have caused the GNN model to ignore possible drug-symptom pairs that would otherwise have been predicted. This illustrates very well the importance and difficulty of finding the right balance between adding concepts to improve adherence to domain knowledge and removing some concepts to hold onto the scope that is actually needed to reach the task objective, which is one of the ideas this project is trying to convey. The fact that such an error was found in the restructured conceptual model, despite having gone through the design approach, may indicate the need for a more careful implementation of this approach, or the need to improve these design steps to enforce this idea of balance between adherence to domain knowledge and necessary scope even more.

To return to the mentioned information distance that might have increased for some connections between concepts after restructuring the knowledge graph, it can be argued that this might also affect the outcome of explanation generation by the GNNExplainer. Placing the concept *gene product* between *drug* and *gene* concepts appears to have decreased the importance of these drug targeting relationships considered by the GNNExplainer as they are not observed in any of the explanations generated from the restructured knowledge graph. This decrease in importance for the drug repurposing link predictions has led to the exclusive appearance of the relation *is substance that treats* in the explanations generated from the restructured knowledge graph while these edges have not been observed in the set of explanations on the original input. This edge type connects a *drug* with a *phenotype*. The presence of this relation might explain the increase in average shortest path length between the drug candidate and the treated symptom since the concept *phenotype* is added to the explaining path followed by gene-gene interactions. For the explanations from the original knowledge graph, the explanation path transitions from a *drug* targeting a *gene* directly towards the gene-gene interactions. This conceptual alteration in the explaining paths requires further investigation to determine whether they provide relevant and useful information to justify the increased complexity.

Based on the analysis of the features of the original and restructured knowledge graphs, the changes in the conceptual structure of the graph caused a decrease in number of relations, a drastic reduction in triples and an increase in total of concepts. Despite these changes, the resulting explanations on the restructured knowledge graph show more conceptual enrichment in each aspect being the average number of node- and edge types found in the explanations as well as the average number of triples. This indicates that restructuring of the input knowledge graph results in an overall higher conceptual enrichment of the explanations. As has been mentioned before with the addition of the concepts *gene product* and *phenotype*, restructuring the model has led to a more fine-grained conceptual structure. This may have caused the users to have perceived the explanations to be more difficult given the lower average ratings on the explainability aspects. Based on the comments given by the participants of the survey, the lack of background working with graphs and the attached description not exactly matching these graphs had a negative impact on understanding the explanations impeding them to assess on the actual content and conceptual structure of the explanation.

Regarding the fine-grained conceptual structure added to the explanations generated from the restructured knowledge graph, it could be suggested that this had some positive effect on the explainability of the explanation. This is based on the comment of one participant who had difficulty in interpreting the meaning of a concept in the explanation on the original knowledge graph. To be specific, the participant felt that the concept *DISO* was ambiguous. In the restructured knowledge graph and thus also in the explanations on this input, the ambiguous concept has been split into the concepts *disease* and *phenotype* which might have solved this observed issue.

During this project, we encountered some limitations of the GNNExplainer as this explanation generation method does not yield a consistent number of explanations per independent run since the algorithm solves an optimization problem in a probabilistic manner as discussed in Section 3.3.6. Ideally, we compare explanations from both knowledge graphs explaining the same drug-symptom pairs. However, the GNNExplainer did not yield subgraphs that explain the same pairs from both input variations. This rules out the possibility to conduct more extensive comparisons that would offer stronger evidence for variations in conceptual enrichment and information paths between the two sets of explanations.

Lastly, the time constraint of this project together with the specific requirements for participating in the survey resulted in a lower number of participants than desired. A greater number of respondents could have led to a different conclusion as it may have resulted in significant differences in the received average ratings.

5.2 Future Work

Although an insignificant difference has been found between the explainability of the XAI pipeline using the original and restructured knowledge graph, we can still discuss whether it would be interesting to invest future research projects on the topic of improved conceptual modelling and XAI.

The created conceptual modelling methods did prove to help verify that the knowledge graph built from various data sources still complies to the relevant domain knowledge and whether the included concepts and relations are useful for reaching the objective of the task. It has been found that expressing conceptual models using a foundational ontology in combination with goal modelling helped communicate the meanings of concepts and relations to experts improving their ability to help validate the designed model. This is why it would be interesting to develop further on finding a way to create conceptual models in a consistent way that helps reach consensus among a group of experts about the represented domain knowledge while also focusing on aligning the concepts and relations such that the model can actually help achieving the objective in its application.

The difficulty of interpreting the explanations generated from the XAI drug repurposing pipeline is shown to be prevalent among the participants of our survey which highlights the fact that explainability in AI is a very complex matter and requires more extensive research and development. However, it is not only about improving the generation of the explanation, but it is equally important to think about how these explanations are represented to the users whose backgrounds often do not align to the developers of these systems. This makes explanations in the form of subgraphs of the input knowledge graph already obsolete when it comes to making it understandable for the decision makers who might consult a XAI system. This means that it needs to be investigated how these explanations can be conveyed in a more user-centric manner such that the representation is compatible and adaptable to the knowledge of the users. For future work, it might be interesting to accompany the developed conceptual modelling approach with a system that enables a more dynamic way of representing explanations allowing the user to choose what information needs to be added and how the representation is formatted such that the user is supported in the understanding of the explanations making use of their own background.

A less drastic step towards a new research project would be to look into other node embedding methods that complement the restructured conceptual structure of the model by replacing the edge2vec method with methods such as metapath2vec [17]. Instead of constructing the neighbourhoods of nodes with random walks constrained by edge types, the metapath2vec model utilizes metapath[1]-based random walks such that the representation learning algorithm yields embeddings retaining the heterogeneity in the node types as well as the edge types.

[1] Metapaths capture the patterns of relations between entities as they are ordered sequences of node- and edge types that are allowed and expected to appear in the knowledge graph.

5.3 Conclusion

This project has developed a hybrid approach for designing a conceptual model that takes into account the accuracy of representing the relevant domain and the scope required to achieve the objective of the task. This conceptual modelling process has been implemented to restructure the knowledge graph that serves as input of a pipeline that obtains drug candidates for DMD-related symptoms developed in the previous XAI-DMD-DR project [57]. The most important aspect of this design process is the use of a foundational ontology that helps express the conceptual model in a clear and precise way. To assess whether the explainability of the generated explanations has improved as a result of our added conceptual modelling approach, we developed a method to measure the explainability in the form of a questionnaire filled in by experts in the drug repurposing and DMD research field. We quantified explainability by asking participants of the survey to rate the more concrete and already established [49] aspects that together cover the term explainability being clarity, parsimony, completeness and soundness.

No significant differences in explainability were found between the explanations generated using the original knowledge graph and the explanations from the restructured knowledge graph. The project started with the research question whether the use of foundational ontologies improves the explainability of XAI algorithms. Based on the acquired results, no significant evidence has been found that shows there was an improvement in the explainability of the XAI drug repurposing algorithm applied on the rare disease DMD using our developed conceptual model design approach.

Appendix A

Models in OntoUML

The higher-quality versions of the OntoUML schemas in Figures A.1, A.2 and A.3 can be accessed at `https://github.com/rosazwart/XAIFO-ThesisProject/tree/main/images`.

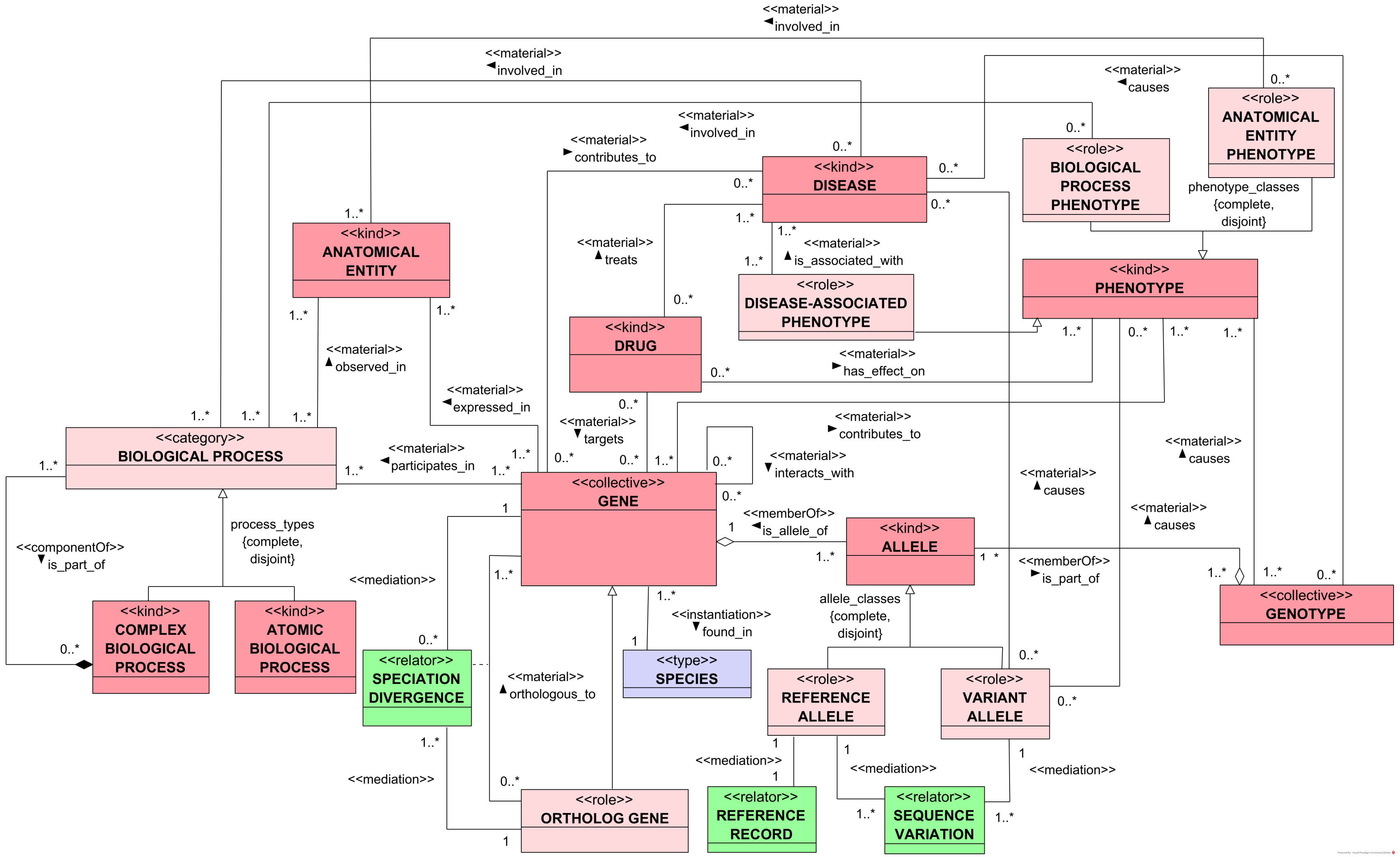

<<category>> BIOLOGICAL PROCESS
<<kind>> ANATOMICAL ENTITY
<<kind>> DISEASE
<<kind>> DRUG
<<role>> DISEASE-ASSOCIATED PHENOTYPE
<<role>> BIOLOGICAL PROCESS PHENOTYPE
<<role>> ANATOMICAL ENTITY PHENOTYPE
<<kind>> PHENOTYPE
phenotype_classes {complete, disjoint}
<<collective>> GENE
<<kind>> ALLELE
<<collective>> GENOTYPE
<<kind>> COMPLEX BIOLOGICAL PROCESS
<<kind>> ATOMIC BIOLOGICAL PROCESS
process_types {complete, disjoint}
<<relator>> SPECIATION DIVERGENCE
<<type>> SPECIES
<<role>> REFERENCE ALLELE
<<role>> VARIANT ALLELE
allele_classes {complete, disjoint}
<<relator>> REFERENCE RECORD
<<relator>> SEQUENCE VARIATION
<<role>> ORTHOLOG GENE
<<material>> involved_in
<<material>> involved_in
<<material>> involved_in
<<material>> causes
<<material>> contributes_to
<<material>> treats
<<material>> is_associated_with
<<material>> observed_in
<<material>> expressed_in
<<material>> has_effect_on
<<material>> targets
<<material>> participates_in
<<material>> contributes_to
<<material>> interacts_with
<<material>> causes
<<material>> causes
<<material>> causes
<<componentOf>> is_part_of
<<memberOf>> is_allele_of
<<memberOf>> is_part_of
<<instantiation>> found_in
<<material>> orthologous_to
<<mediation>>
<<mediation>>
<<mediation>>
<<mediation>>
<<mediation>>

Figure A.1: The OntoUML schema expressing the *domain model* resulting from the top-down approach.

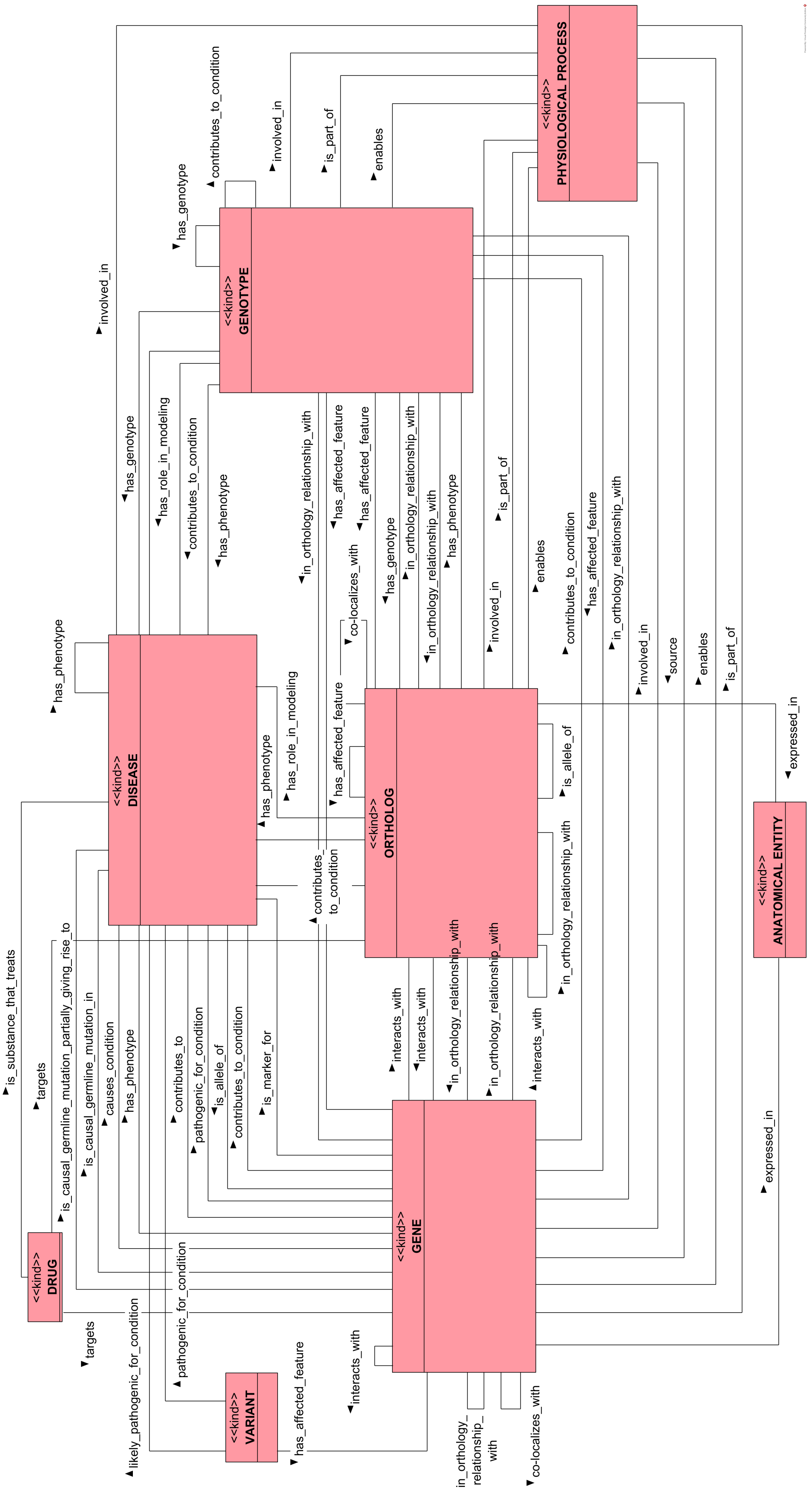
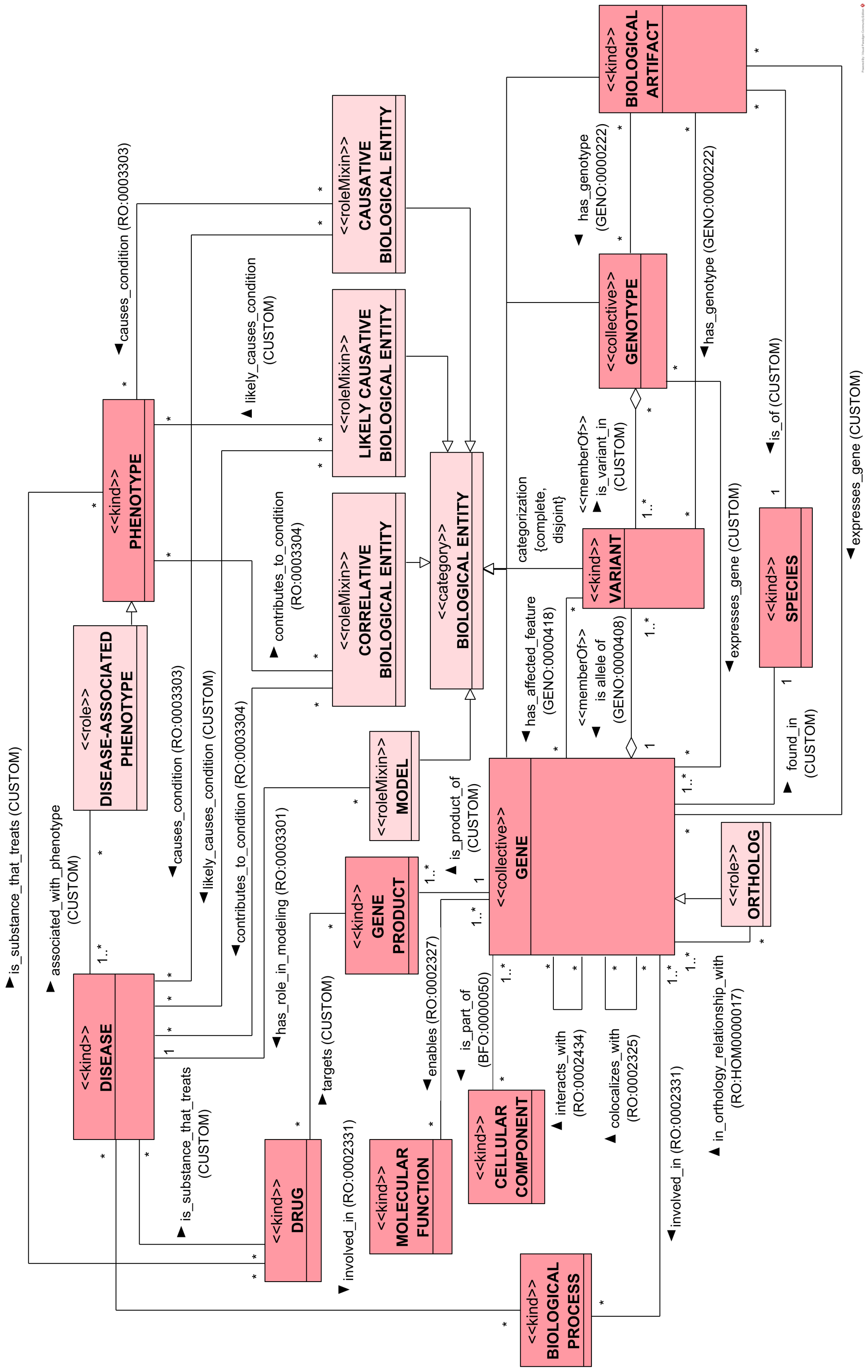


Figure A.2: The OntoUML schema expressing the *data-based model* resulting from the bottom-up approach.



<<kind>> BIOLOGICAL ARTIFACT
<<roleMixin>> CAUSATIVE BIOLOGICAL ENTITY
<<roleMixin>> LIKELY CAUSATIVE BIOLOGICAL ENTITY
<<roleMixin>> CORRELATIVE BIOLOGICAL ENTITY
<<category>> BIOLOGICAL ENTITY
<<roleMixin>> MODEL
<<kind>> GENE PRODUCT
<<collective>> GENE
<<collective>> GENOTYPE
<<kind>> VARIANT
<<kind>> SPECIES
<<role>> ORTHOLOG
<<kind>> PHENOTYPE
<<role>> DISEASE-ASSOCIATED PHENOTYPE
<<kind>> DISEASE
<<kind>> DRUG
<<kind>> MOLECULAR FUNCTION
<<kind>> CELLULAR COMPONENT
<<kind>> BIOLOGICAL PROCESS
◄causes_condition (RO:0003303)
▲ likely_causes_condition (CUSTOM)
▲ contributes_to_condition (RO:0003304)
◄causes_condition (RO:0003303)
◄likely_causes_condition (CUSTOM)
◄contributes_to_condition (RO:0003304)
◄has_role_in_modeling (RO:0003301)
▲ associated_with_phenotype (CUSTOM)
▲is_substance_that_treats (CUSTOM)
▲ is_substance_that_treats (CUSTOM)
◄ targets (CUSTOM)
◄ involved_in (RO:0002331)
◄enables (RO:0002327)
◄ is_part_of (BFO:0000050)
▲ is_product_of (CUSTOM)
▲ interacts_with (RO:0002434)
▲ colocalizes_with (RO:0002325)
◄involved_in (RO:0002331)
▲ in_orthology_relationship_with (RO:HOM0000017)
◄ has_affected_feature (GENO:0000418)
<<memberOf>> ◄ is allele of (GENO:0000408)
categorization {complete, disjoint}
<<memberOf>> ▲ is_variant_in (CUSTOM)
◄ has_genotype (GENO:0000222)
◄has_genotype (GENO:0000222)
◄expresses_gene (CUSTOM)
◄is_of (CUSTOM)
▲ found_in (CUSTOM)
◄expresses_gene (CUSTOM)
*
1
1..*

Figure A.3: The OntoUML schema expressing the *intermediary conceptual model* resulting from the hybrid conceptual modelling method.

Appendix B

Overview Encountered Relations from OBO Relations Ontology and GENO Ontology

Table B.1: This table contains the relations present in the data instances and other relevant relations with their identifiers and definitions. The relations and their annotations have been found in the OBO Relations Ontology [52] or the GENO ontology [12]. The relations originate from the OBO Relations Ontology when the identifier starts with “RO”. Relations come from the GENO ontology that have the prefix “GENO” in their identifiers. All definitions are taken directly from the ontologies.

Relation	Identifier	Definition
has affected feature	GENO:0000418	A relation that holds between an instance of a genetic variation and a genomic feature (typically a gene class) that is affected in its sequence or expression.

Continued on next page

Table B.1 – continued from previous page

Relation	Identifier	Definition
is allele of	GENO:0000408	A relation linking an instance of a variable feature (aka an allele) to a genomic location/locus it occupies. This is typically a gene locus, but a feature may be an allele of other types of named loci such as QTLs, or alleles of some unnamed locus of arbitrary size.
is expression variant of	GENO:0000443	A relation between an expression-variant gene (ie integrated transgenes or knockdown reagent targeted genes), and the class of gene it represents.
targets gene	GENO:0000414	A relation between a gene targeting reagent (e.g. a morpholino or RNAi) and the class of gene it targets.
in orthology relationship with	RO:HOM0000017	Historical homology that involves genes that diverged after a speciation event.
in 1 to 1 orthology relationship with	RO:HOM0000020	Orthology that involves two genes that did not experience any duplication after the speciation event that created them.
contributes to condition	RO:0003304	A relationship between an entity (e.g. a genotype, genetic variation, chemical, or environmental exposure) and a condition (a phenotype or disease), where the entity has some contributing role that influences the condition.

Continued on next page

Table B.1 – continued from previous page

Relation	Identifier	Definition
causes condition	RO:0003303	A relationship between an entity (e.g. a genotype, genetic variation, chemical, or environmental exposure) and a condition (a phenotype or disease), where the entity has some causal role for the condition.
is causal germline mutation partially giving rise to	RO:0004016	Relates a gene to condition, such that a mutation in this gene partially contributes to the presentation of this condition.
is causal germline mutation in	RO:0004013	Relates a gene to condition, such that a mutation in this gene is sufficient to produce the condition and that can be passed on to offspring.
pathogenic for condition	GENO:0000840	-
likely pathogenic for condition	GENO:0000841	-
is marker for	RO:0002607	c is marker for d iff the presence or occurrence of d is correlated with the presence of occurrence of c, and the observation of c is used to infer the presence or occurrence of d. Note that this does not imply that c and d are in a direct causal relationship, as it may be the case that there is a third entity e that stands in a direct causal relationship with c and d.
involved in	RO:0002331	c involved in p if and only if c enables some process p', and p' is part of p

Continued on next page

Table B.1 – continued from previous page

Relation	Identifier	Definition
enables	RO:0002327	c enables p iff c is capable of p and c acts to execute p.
is part of	BFO:0000050	A core relation that holds between a part and its whole.
contributes to	RO:0002326	-
has phenotype	RO:0002200	A relationship that holds between a biological entity and a phenotype. Here a phenotype is construed broadly as any kind of quality of an organism part, a collection of these qualities, or a change in quality or qualities (e.g. abnormally increased temperature). The subject of this relationship can be an organism (where the organism has the phenotype, i.e. the qualities inhere in parts of this organism), a genomic entity such as a gene or genotype (if modifications of the gene or the genotype causes the phenotype), or a condition such as a disease (such that if the condition inheres in an organism, then the organism has the phenotype).

Appendix C

Dataset Features

subject	relation	object
ORTH	None	ORTH
GENO	None	ORTH
GENO	None	GENE
DRUG	None	DISO
DISO	None	GENO
GENE	causes condition	DISO
GENE	colocalizes with	GENE
ORTH	colocalizes with	ORTH
GENE	contributes to	DISO
GENO	contributes to condition	GENO
GENO	contributes to condition	DISO
GENE	contributes to condition	GENO
ORTH	contributes to condition	DISO
GENE	contributes to condition	DISO
ORTH	enables	PHYS
GENO	enables	PHYS
GENE	enables	PHYS
ORTH	expressed in	ANAT
GENE	expressed in	ANAT
GENO	has affected feature	ORTH
VARI	has affected feature	GENE
ORTH	has affected feature	ORTH
GENO	has affected feature	GENE
GENO	has genotype	ORTH
GENO	has genotype	GENO
GENO	has genotype	DISO
GENO	has phenotype	DISO
ORTH	has phenotype	DISO
ORTH	has phenotype	GENO
DISO	has phenotype	DISO
GENE	has phenotype	DISO
GENO	has role in modeling	DISO
ORTH	has role in modeling	DISO
GENO	in 1 to 1 orthology relationship with	ORTH
ORTH	in 1 to 1 orthology relationship with	GENO

subject	relation	object
GENE	in 1 to 1 orthology relationship with	GENE
GENE	in 1 to 1 orthology relationship with	GENO
ORTH	in 1 to 1 orthology relationship with	GENE
GENO	in 1 to 1 orthology relationship with	GENE
GENE	in 1 to 1 orthology relationship with	ORTH
ORTH	in 1 to 1 orthology relationship with	ORTH
ORTH	in orthology relationship with	ORTH
ORTH	in orthology relationship with	GENO
GENO	in orthology relationship with	GENE
GENE	in orthology relationship with	ORTH
GENO	in orthology relationship with	ORTH
GENE	in orthology relationship with	GENE
GENE	in orthology relationship with	GENO
ORTH	in orthology relationship with	GENE
GENE	interacts with	ORTH
ORTH	interacts with	GENE
ORTH	interacts with	ORTH
GENE	interacts with	GENE
ORTH	involved in	PHYS
DISO	involved in	PHYS
GENO	involved in	PHYS
GENE	involved in	PHYS
DISO	is allele of	GENE
ORTH	is allele of	ORTH
GENE	is causal germline mutation in	DISO
GENE	is causal germline mutation partially giving rise to	DISO
GENE	is marker for	DISO
ORTH	is part of	PHYS
GENE	is part of	PHYS
GENO	is part of	PHYS
DRUG	is substance that treats	DISO
VARI	likely pathogenic for condition	DISO
GENE	pathogenic for condition	DISO
VARI	pathogenic for condition	DISO
PHYS	source	GENE
DRUG	targets	ORTH
DRUG	targets	GENE

Table C.1: All triples found in the original dataset acquired by using the data fetch scripts from [57]. The table shows the subject, object and relation of each existing triplet. These are also all the triples, totalling 72, found in the original knowledge graph.

subject	relation	object
gene	causes condition	disease
gene	colocalizes with	gene
variant	contributes to condition	phenotype
gene	contributes to condition	phenotype
gene	contributes to condition	disease
variant	contributes to condition	disease
model	enables	molecular function
gene	enables	molecular function

subject	relation	object
gene	expressed in	anatomical entity
variant	has affected feature	gene
model	has genotype	genotype
model	has genotype	variant
model	has phenotype	phenotype
genotype	has phenotype	phenotype
disease	has phenotype	phenotype
variant	has phenotype	phenotype
gene	has phenotype	phenotype
variant	has phenotype	disease
genotype	has phenotype	disease
variant	has role in modeling	disease
model	has role in modeling	disease
genotype	has role in modeling	disease
gene	has role in modeling	disease
gene	in 1 to 1 orthology relationship with	model
gene	in 1 to 1 orthology relationship with	gene
model	in 1 to 1 orthology relationship with	gene
gene	in orthology relationship with	gene
gene	interacts with	model
gene	interacts with	gene
model	interacts with	gene
gene	involved in	pathway
disease	involved in	biological process
gene	involved in	biological process
disease	involved in	pathway
variant	is allele of	gene
gene	is causal germline mutation in	disease
gene	is causal germline mutation partially giving rise to	disease
gene	is part of	cellular component
model	is part of	cellular component
gene product	is product of	gene
drug	is substance that treats	phenotype
variant	likely pathogenic for condition	disease
model	nan	gene
genotype	nan	gene
chemical	nan	disease
variant	nan	genotype
variant	nan	model
variant	pathogenic for condition	disease
drug	targets	gene product

Table C.2: All triples found in the dataset acquired from the newly built fetcher, showing the subject, object and relation of each existing triplet. More details about the relations can be found in Table B.1.

subject	relation	object
disease	associated with phenotype	phenotype
genotype	causes condition	disease
biological artifact	causes condition	phenotype
gene	causes condition	phenotype

subject	**relation**	**object**
variant	causes condition	phenotype
genotype	causes condition	phenotype
gene	causes condition	disease
variant	causes condition	disease
gene	colocalizes with	gene
variant	contributes to condition	disease
gene	contributes to condition	disease
variant	contributes to condition	phenotype
gene	contributes to condition	phenotype
gene	enables	molecular function
genotype	expresses gene	gene
biological artifact	expresses gene	gene
gene	found in	taxon
variant	has affected feature	gene
biological artifact	has genotype	variant
biological artifact	has genotype	genotype
variant	has role in modeling	disease
gene	has role in modeling	disease
genotype	has role in modeling	disease
biological artifact	has role in modeling	disease
gene	in orthology relationship with	gene
gene	interacts with	gene
disease	involved in	biological process
gene	involved in	biological process
variant	is allele of	gene
biological artifact	is of	taxon
gene	is part of	cellular component
gene product	is product of	gene
drug	is substance that treats	disease
drug	is substance that treats	phenotype
variant	is variant in	genotype
variant	likely causes condition	disease
drug	targets	gene product

Table C.3: All 37 triples found in the restructured knowledge graph.

Appendix D

Explanations

D.1 Using Original Knowledge Graph

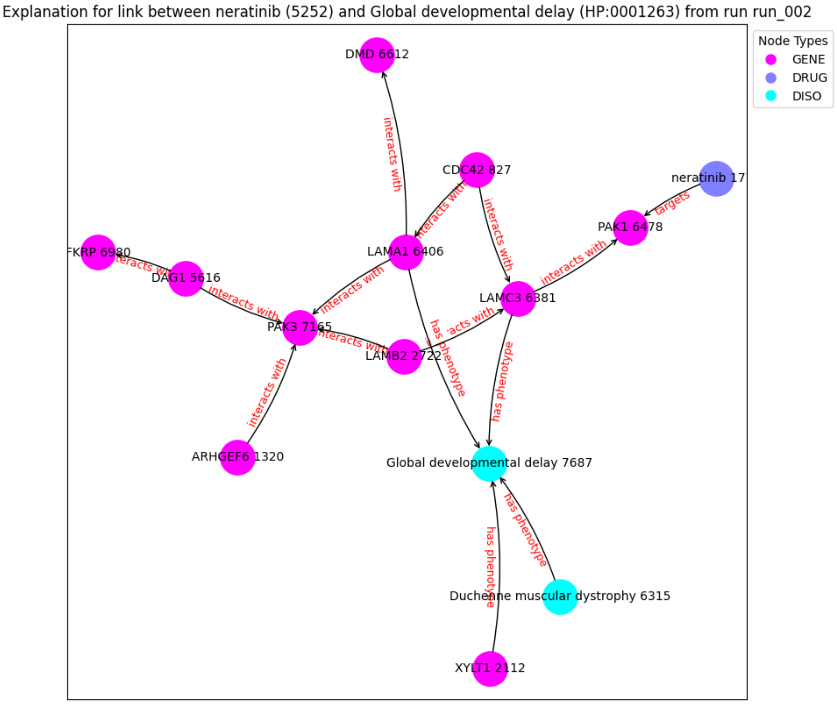


Figure D.1: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating global developmental delay.

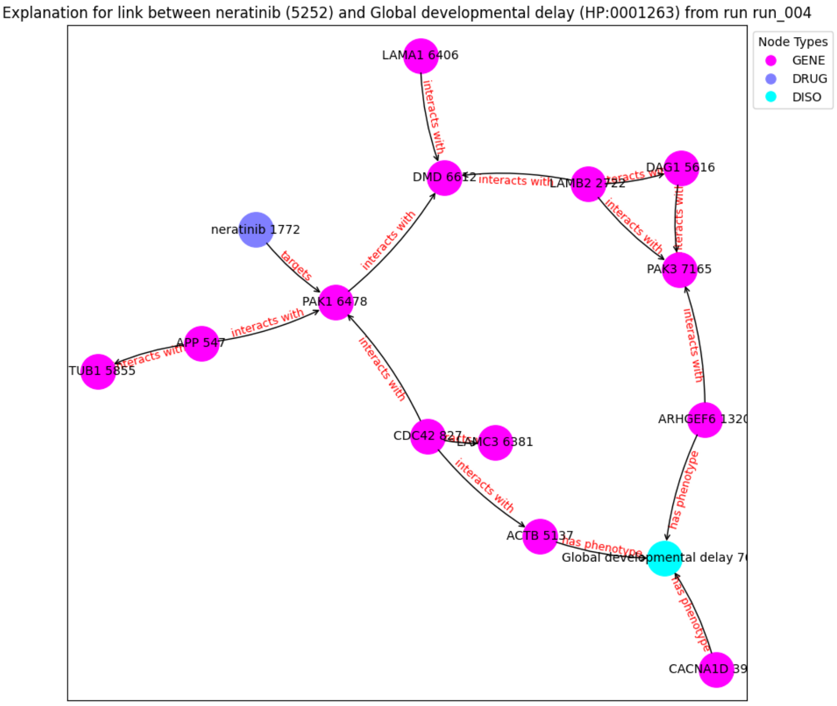


Figure D.2: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating global developmental delay.

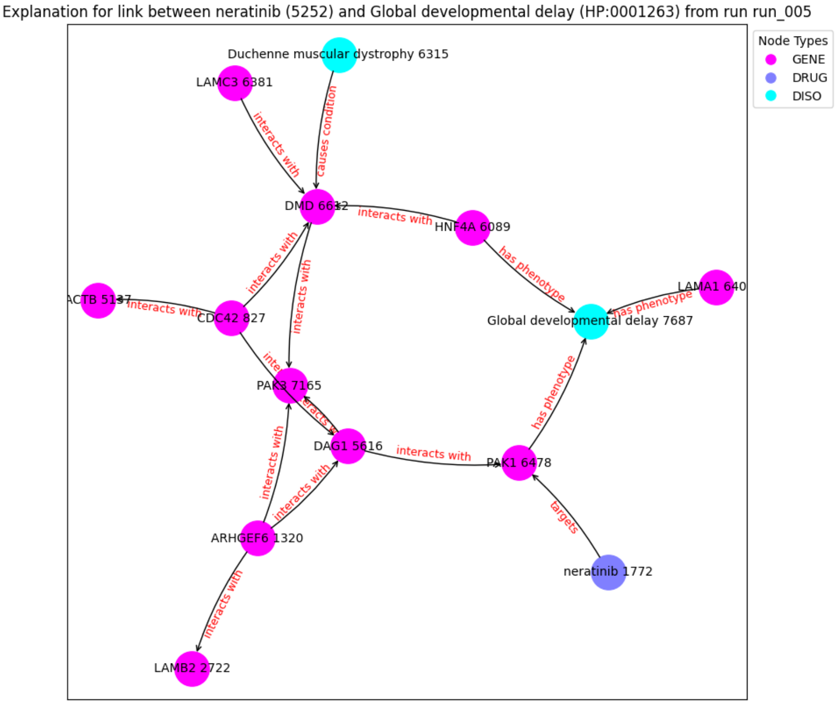


Figure D.3: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating global developmental delay. This explanation is added to Questionnaire 2 (see questionnaires in Appendix E.1).

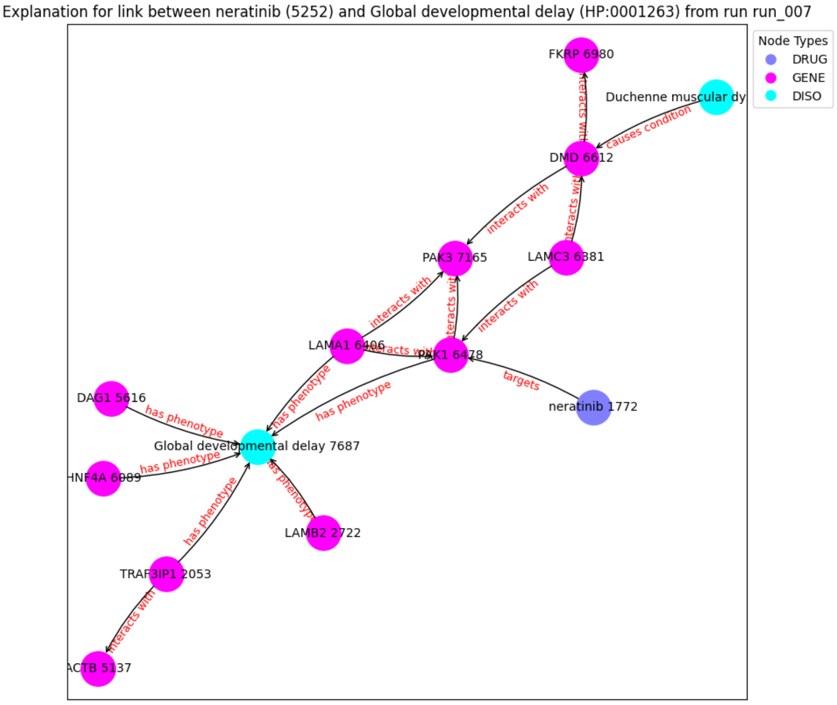


Figure D.4: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating global developmental delay.

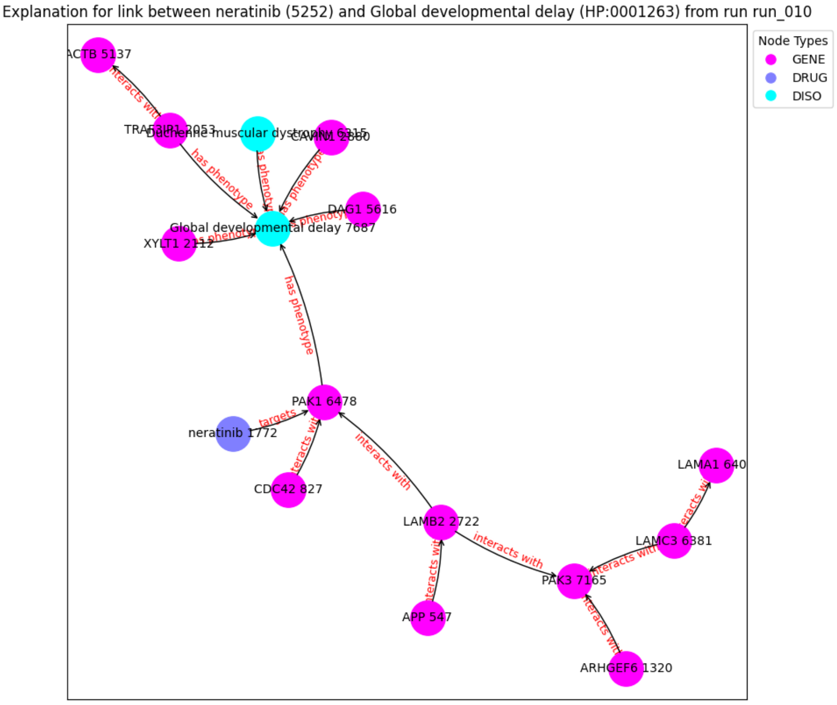


Figure D.5: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating global developmental delay.

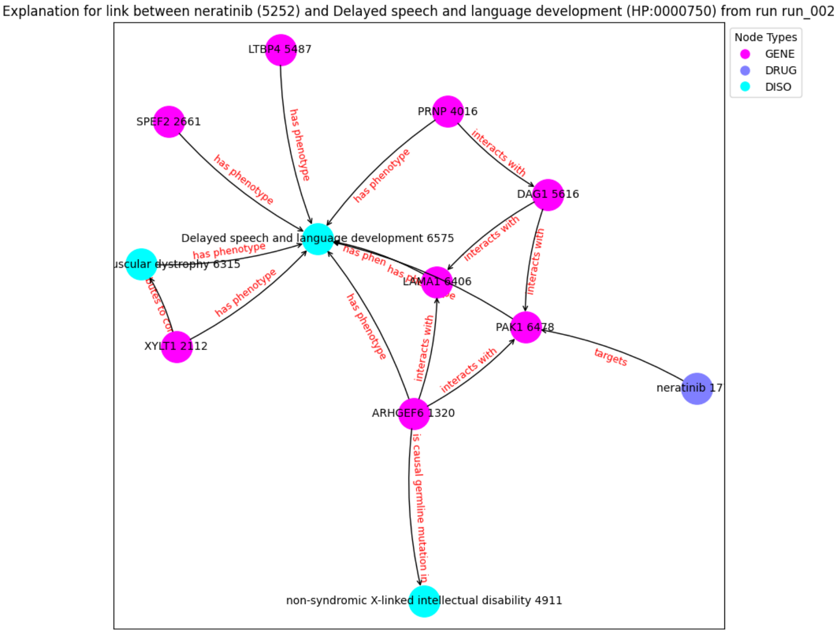

Explanation for link between neratinib (5252) and Delayed speech and language development (HP:0000750) from run run_002
Node Types
GENE
DRUG
DISO
LTBP4 5487
has phenotype
SPEF2 2661
PRNP 4016
interacts with
DAG1 5616
Delayed speech and language development 6575
uscular dystrophy 6315
LAMA1 6406
PAK1 6478
XYLT1 2112
targets
neratinib 17
ARHGEF6 1320
is causal germline mutation in
non-syndromic X-linked intellectual disability 4911

Figure D.6: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating delayed speech and language development.

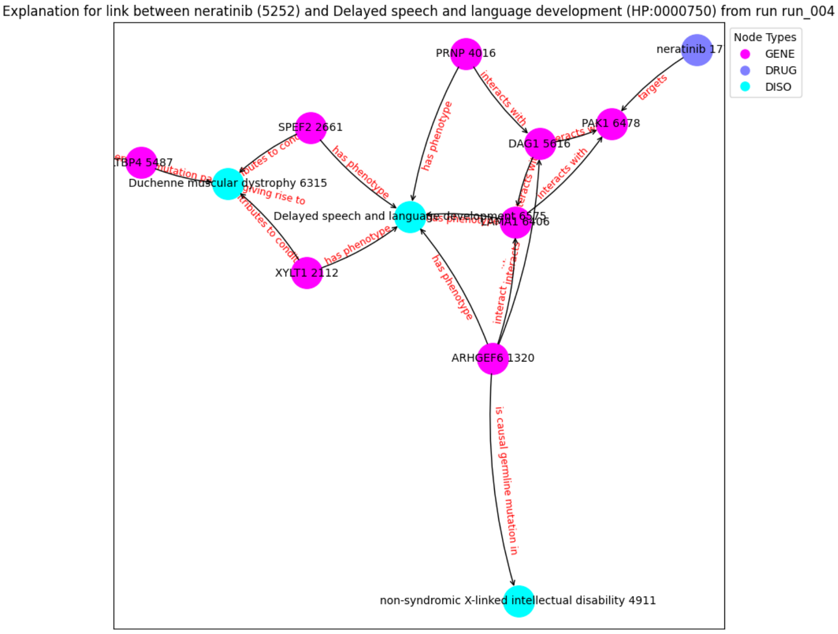

Explanation for link between neratinib (5252) and Delayed speech and language development (HP:0000750) from run run_004
Node Types
GENE
DRUG
DISO
PRNP 4016
neratinib 17
targets
interacts with
has phenotype
PAK1 6478
DAG1 5616
SPEF2 2661
LTBP4 5487
Duchenne muscular dystrophy 6315
giving rise to
has phenotype
Delayed speech and language development 6575
LAMA1 6406
interacts with
XYLT1 2112
has phenotype
has phenotype
interact
ARHGEF6 1320
is causal germline mutation in
non-syndromic X-linked intellectual disability 4911

Figure D.7: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating delayed speech and language development.

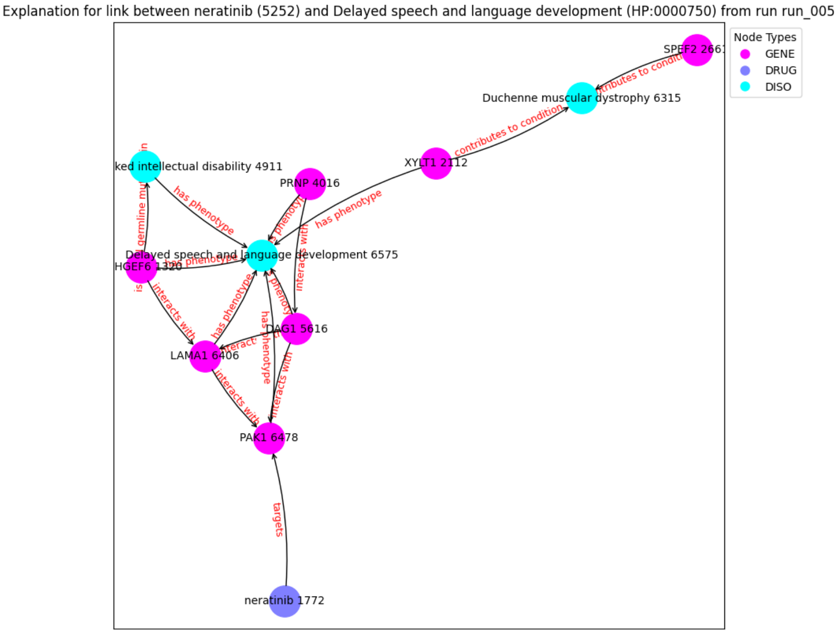


Figure D.8: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating delayed speech and language development.

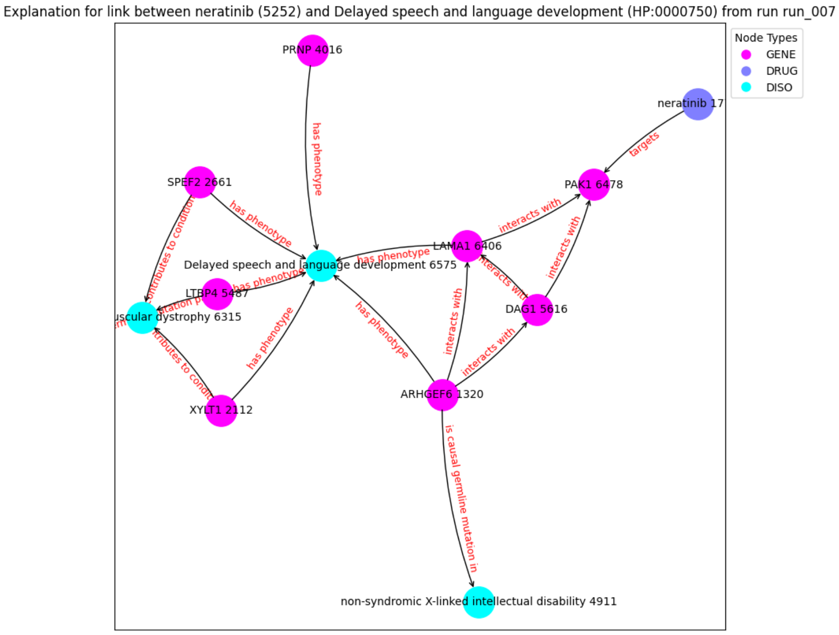


Figure D.9: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating delayed speech and language development. This explanation is added to Questionnaire 1 (see questionnaires in Appendix E.1).

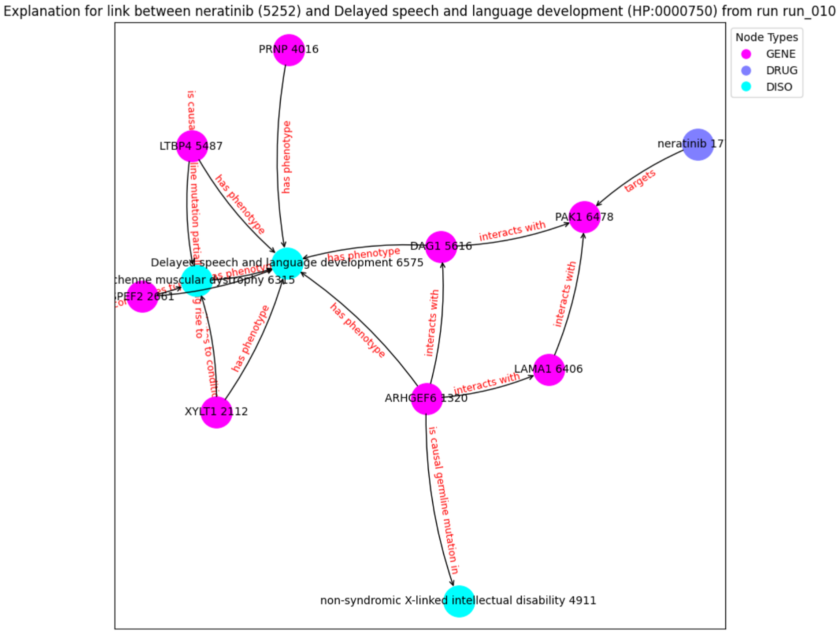

Explanation for link between neratinib (5252) and Delayed speech and language development (HP:0000750) from run run_010
Node Types
GENE
DRUG
DISO
PRNP 4016
is causal germline mutation partially
LTBP4 5487
has phenotype
has phenotype
neratinib 17
targets
PAK1 6478
interacts with
DAG1 5616
has phenotype
Delayed speech and language development 6575
chenne muscular dystrophy 6315
has phenotype
SPEF2 2661
giving rise to
contributes to condition
has phenotype
has phenotype
interacts with
interacts with
LAMA1 6406
interacts with
ARHGEF6 1320
XYLT1 2112
is causal germline mutation in
non-syndromic X-linked intellectual disability 4911

Figure D.10: Generated explanation given the model trained on the original knowledge graph. The subgraph of the input knowledge graph shows the explanation that neratinib is a possible drug candidate for treating delayed speech and language development.

D.2 Using Restructured Knowledge Graph

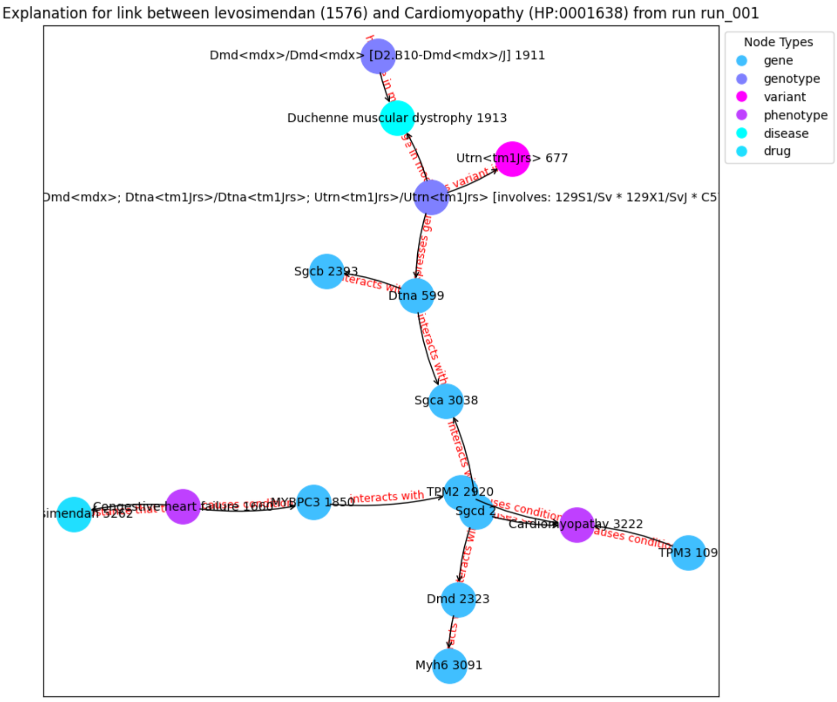

Explanation for link between levosimendan (1576) and Cardiomyopathy (HP:0001638) from run run_001
Node Types
gene
genotype
variant
phenotype
disease
drug
Dmd<mdx>/Dmd<mdx> [D2.B10-Dmd<mdx>/J] 1911
Duchenne muscular dystrophy 1913
Utrn<tm1Jrs> 677
variant
Dmd<mdx>; Dtna<tm1Jrs>/Dtna<tm1Jrs>; Utrn<tm1Jrs>/Utrn<tm1Jrs> [involves: 129S1/Sv * 129X1/SvJ * C5
presses ge
Sgcb 2393
interacts wi
Dtna 599
interacts with
Sgca 3038
interacts
imendan 5262
Congestive heart failure 1660
causes condition
MYBPC3 1850
interacts with
TPM2 2920
Sgcd 2
causes condition
Cardiomyopathy 3222
causes conditio
TPM3 109
Dmd 2323
Myh6 3091

Figure D.11: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy.

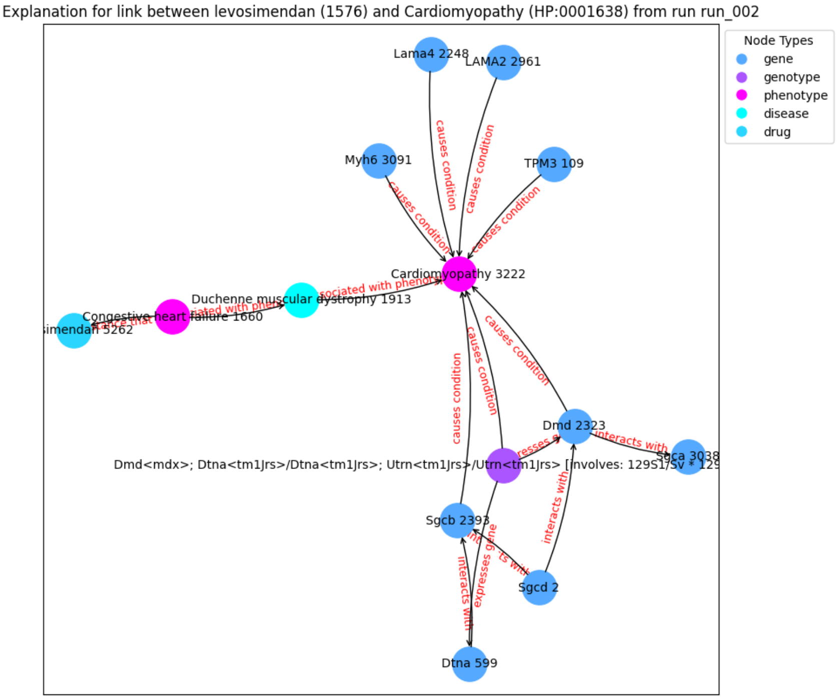


Figure D.12: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy.

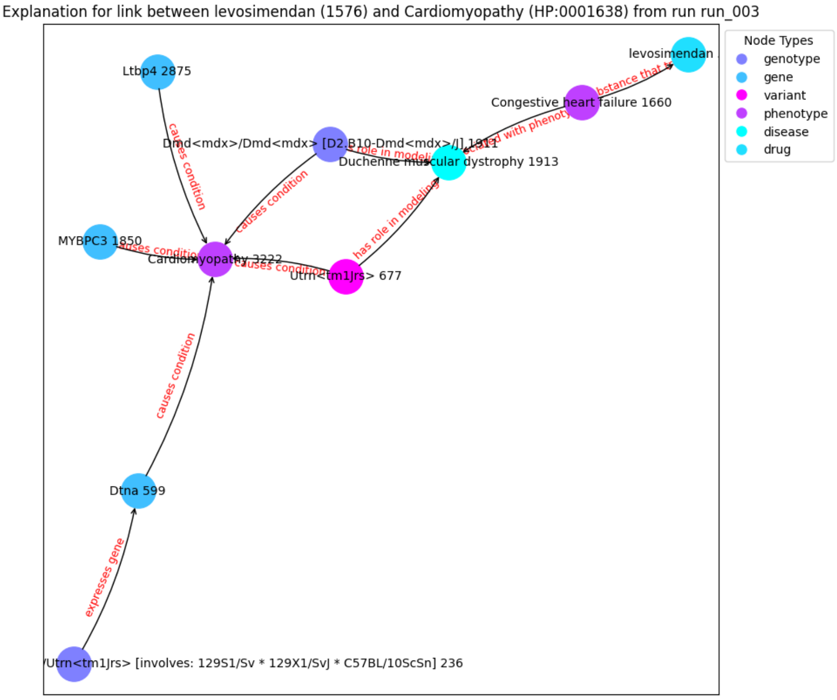


Figure D.13: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy.

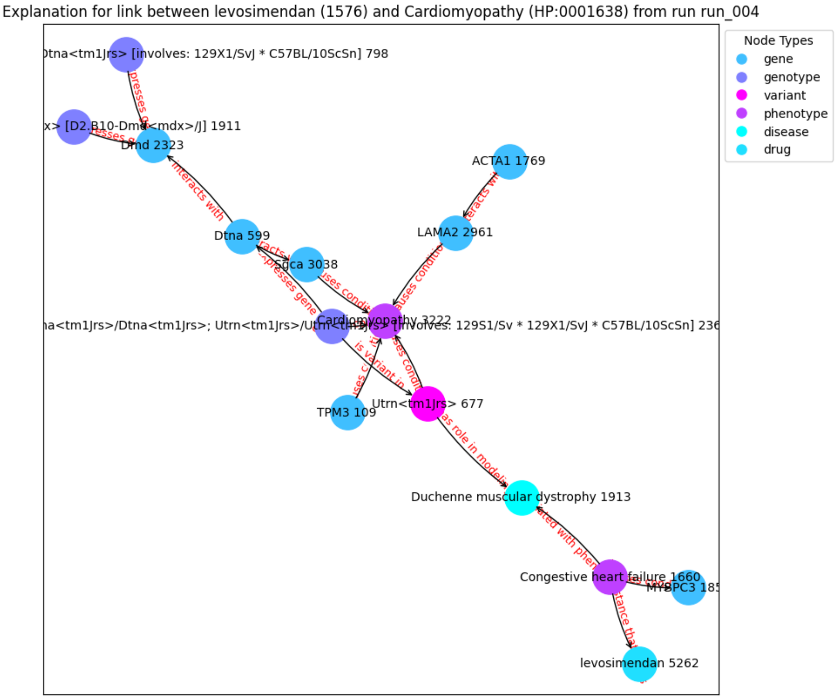

Explanation for link between levosimendan (1576) and Cardiomyopathy (HP:0001638) from run run_004
Node Types
gene
genotype
variant
phenotype
disease
drug
)tna<tm1Jrs> [involves: 129X1/SvJ * C57BL/10ScSn] 798
x> [D2.B10-Dmd<mdx>/J] 1911
Dmd 2323
ACTA1 1769
LAMA2 2961
Dtna 599
Sgca 3038
Cardiomyopathy 3222
ia<tm1Jrs>/Dtna<tm1Jrs>; Utrn<tm1Jrs>/Utrn<tm1Jrs> [involves: 129S1/Sv * 129X1/SvJ * C57BL/10ScSn] 236
TPM3 109
Utrn<tm1Jrs> 677
Duchenne muscular dystrophy 1913
Congestive heart failure 1660
MYBPC3 185
levosimendan 5262
interacts with
expresses gene
causes condition
is variant in
has role in modeling
associated with phenotype

Figure D.14: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy. This explanation is added to Questionnaire 5 (see questionnaires in Appendix E.1).

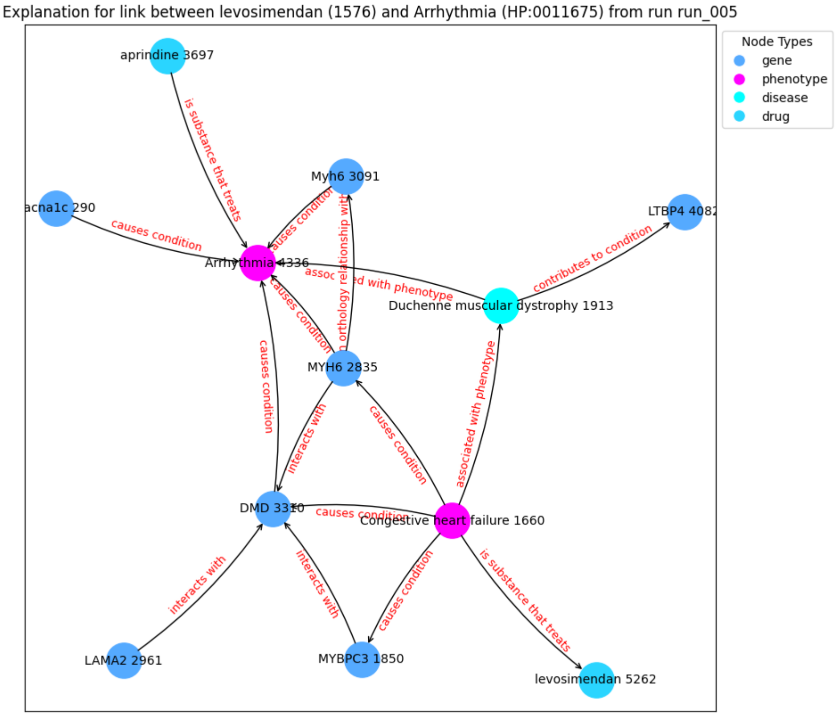
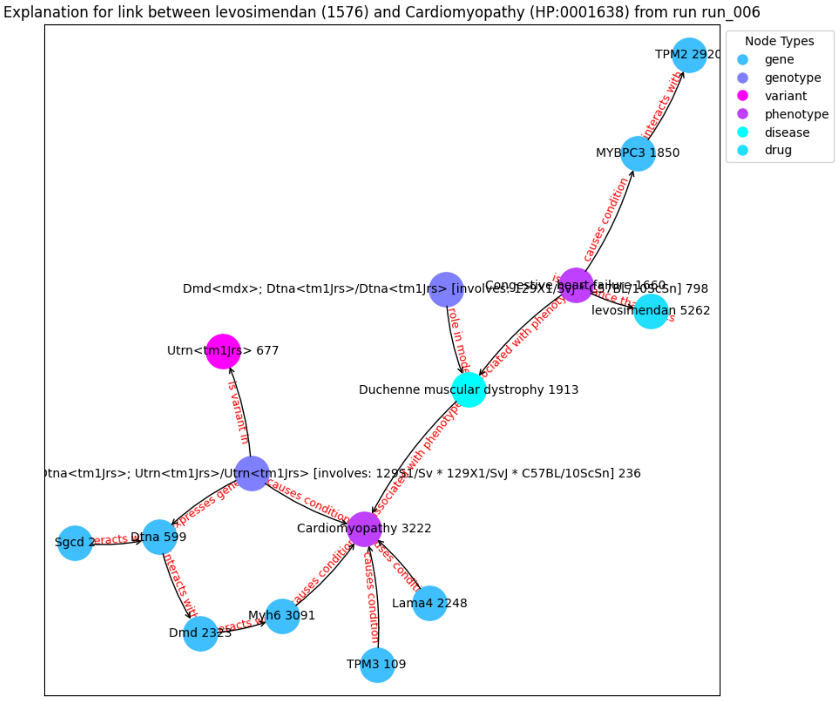


Figure D.15: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.



Explanation for link between levosimendan (1576) and Cardiomyopathy (HP:0001638) from run run_006

Figure D.16: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy.

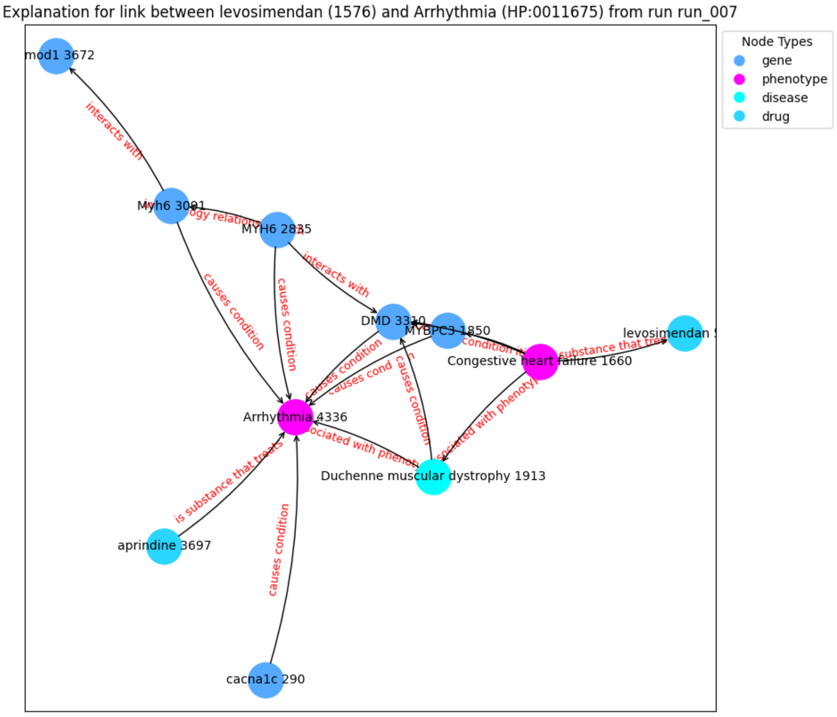


Figure D.17: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

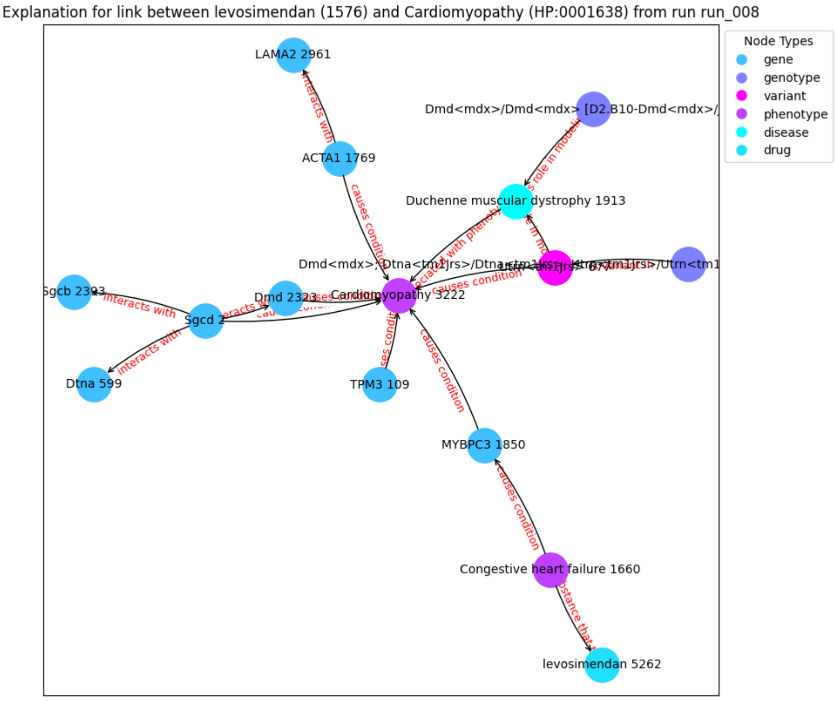


Figure D.18: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy.

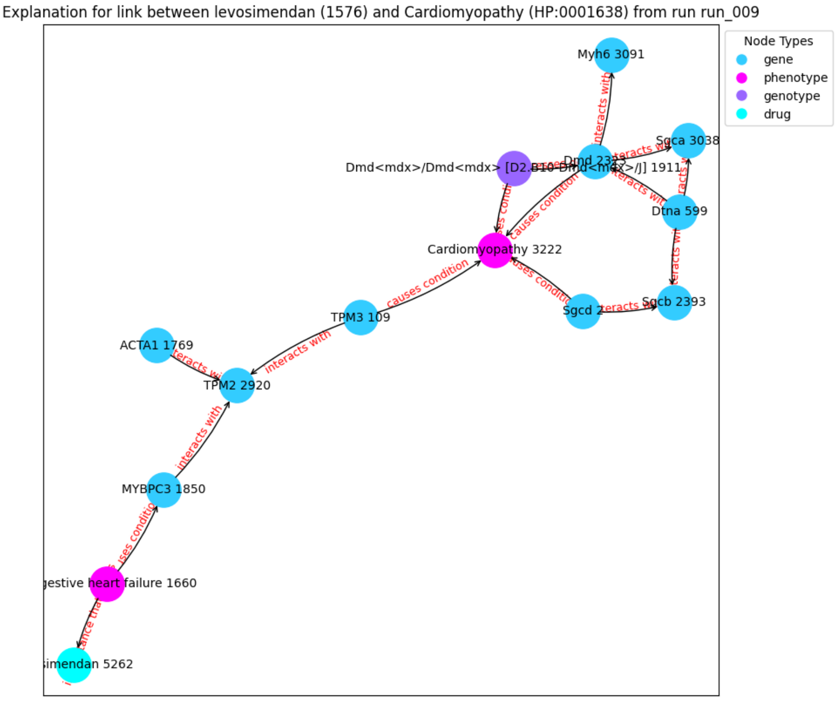


Figure D.19: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating cardiomyopathy.

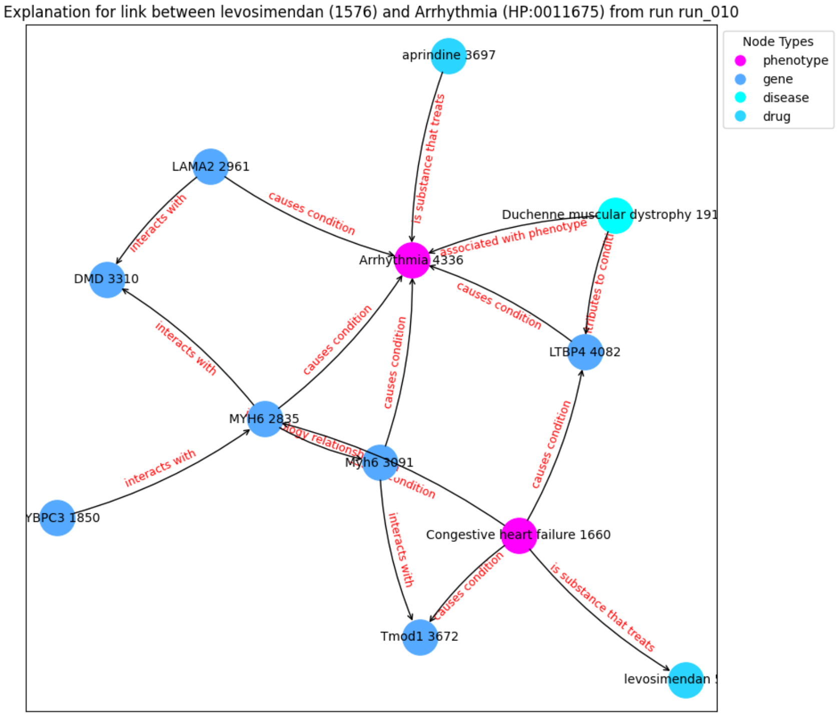


Figure D.20: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

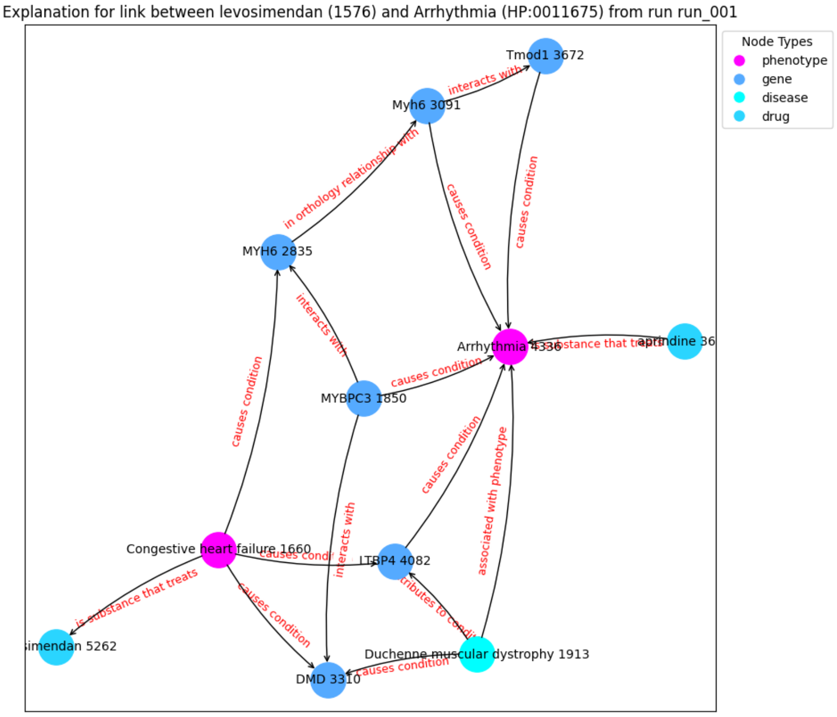


Figure D.21: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

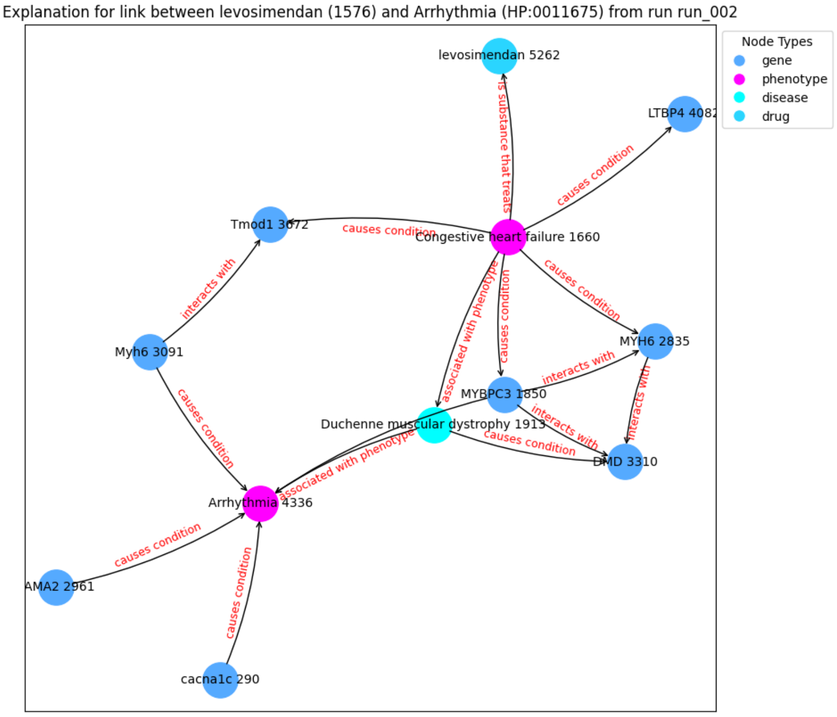


Figure D.22: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

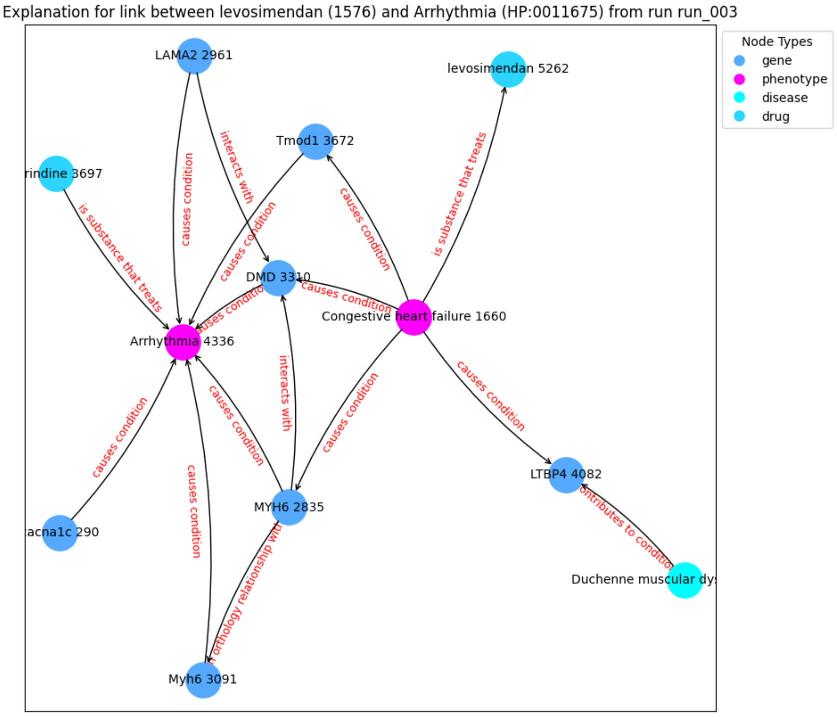


Figure D.23: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

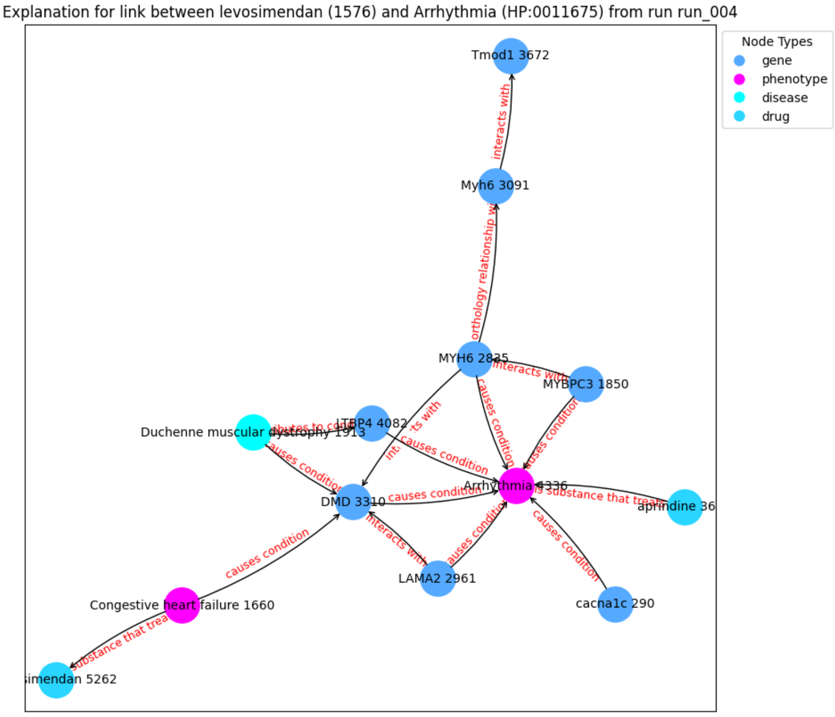


Figure D.24: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

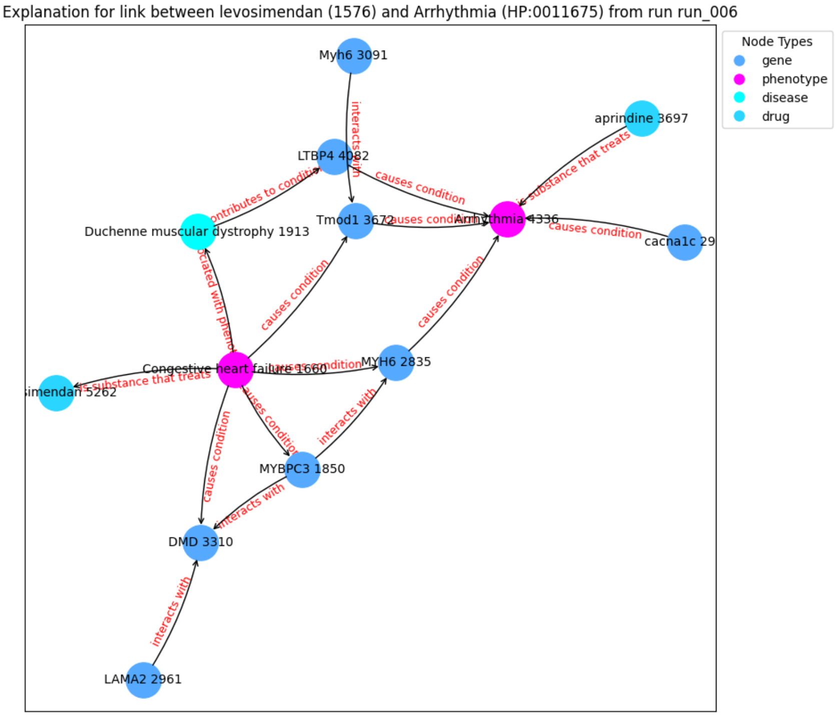


Figure D.25: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia. This explanation is added to Questionnaire 4 (see questionnaires in Appendix E.1).

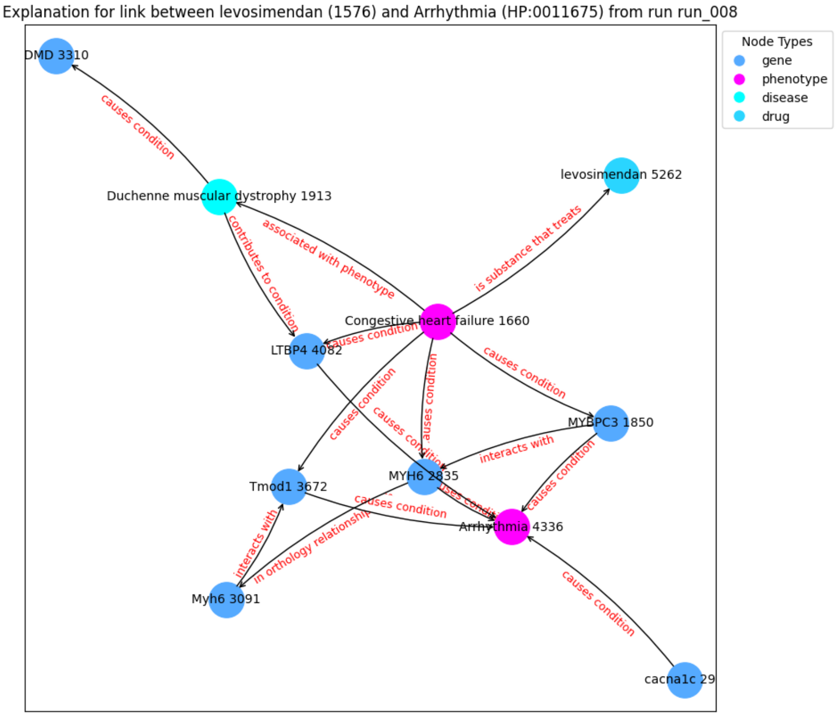


Figure D.26: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

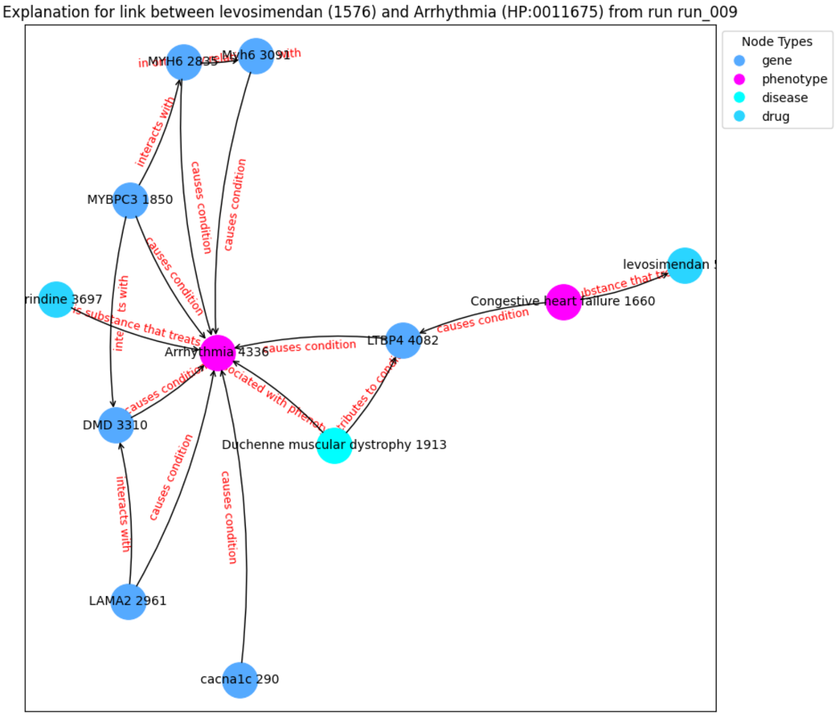


Figure D.27: Generated explanation given the model trained on the restructured knowledge graph. The subgraph of the input knowledge graph shows the explanation that levosimendan is a possible drug candidate for treating arrhythmia.

Appendix E

Questionnaires

E.1 Google Forms

#1 Explainable AI for DMD Drug-Repurposing

Obtaining drug candidates that can treat symptoms observed in Duchenne muscular dystrophy disease

* Indicates required question

This questionnaire is part of a Leiden University master's thesis project performed by Rosa Zwart in collaboration with the LUMC BioSemantics group. This project aims to investigate whether the explainability of explainable AI algorithms can be improved by using Foundational Ontologies.

Introduction

This questionnaire is meant for obtaining feedback from users with knowledge and experience in drug repurposing and/or the rare disease DMD.

A knowledge graph is created containing phenotypic and genetic information directly and indirectly associated with Duchenne muscular dystrophy (DMD). Subsequently, data have been incorporated into the graph that relate to drugs and their associated protein targets as well as information regarding the symptoms and diseases for which a drug is used during treatment. Using this knowledge graph, a deep learning model is trained which results in predictions of new links between drugs and phenotypes directly associated with DMD. Due to the black-box nature of deep learning algorithms, it is difficult for human decision-makers to assess the reliability of these predictions. To enable decision-makers to subject the predictions obtained by the model to their own reasoning, interpretable explanations are added to the link predictions of the trained graph neural network (GNN) model. By adding an explanation-generating method to the deep learning algorithm, we arrive at the field of explainable AI that already shows promising developments into the direction of applying trustworthy deep learning techniques to important medical decisions.

During this research project, we want to investigate whether the explanations generated for the predictions by the trained model can be improved. To attempt this, we tested two different approaches, and we are now running a blind assessment of the results. In this questionnaire, you got an explanation generated by one of the two approaches.

The explanations are subgraphs that explain the predicted existence of a certain link between a drug and DMD-related symptom node by showing the most impactful links that lead to this prediction.

Background Information

In order to acquire an idea about the background of all participants, some questions will be asked regarding your research experience.

1. How many years of experience do you have in the field of drug repurposing and/or rare diseases research? *

____

2. Do you conduct research on the rare disease Duchenne muscular dystrophy? *

Mark only one oval.

○ Yes

○ No

Explanation

To assess the explainability of the given explanation, Likert scaled scores need to be assigned for each explainability aspect.

A caption is added to the subgraph to show the meaning of the most relevant paths for explaining the prediction and to provide additional information about the instances found in the graphs.

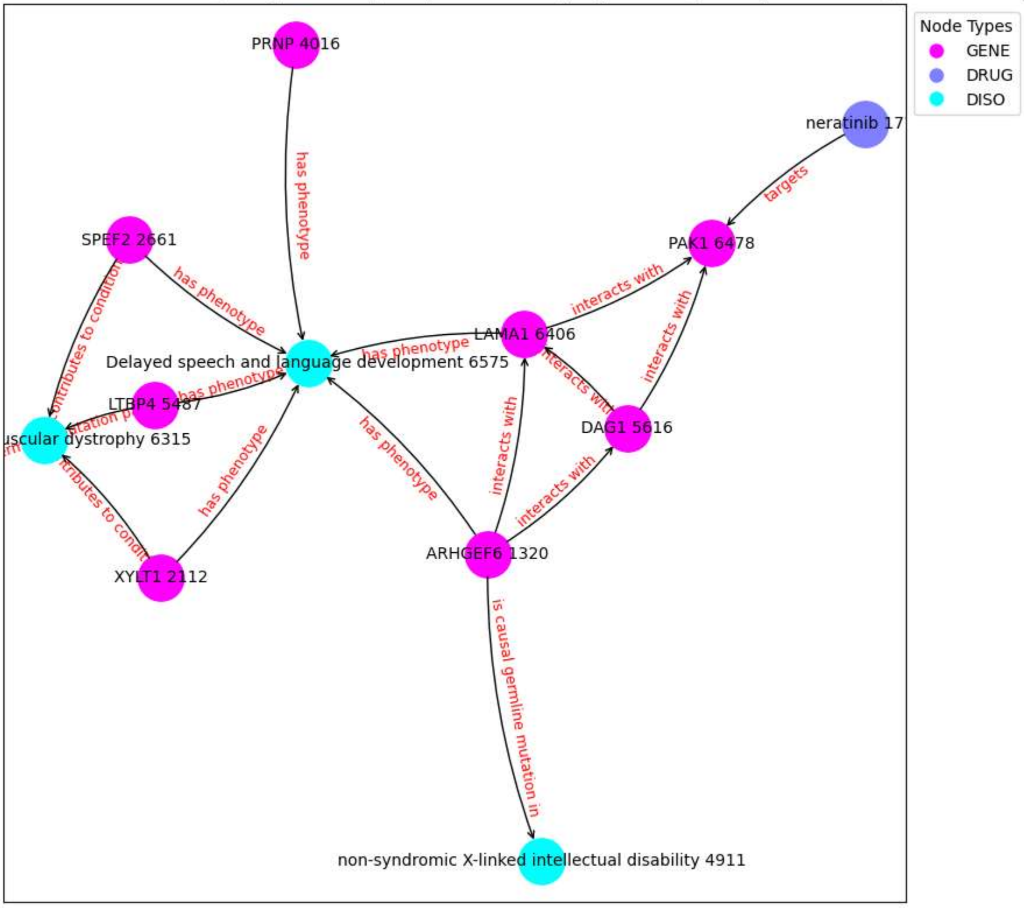
Prediction

It has been predicted that neratinib treats delayed speech and language development.

Explanation (Check the graph before reading the text below)

Neratinib targets the gene PAK1. PAK1 interacts with genes DAG1 and LAMA1. LAMA1 and DAG1 interact with ARHGEF6 and with each other. Mutations in ARHGEF6 and LAMA1 cause the phenotype delayed speech and language development.

Explanation for link between neratinib (5252) and Delayed speech and language development (HP:0000750) from run run_007



3. **Clarity** *

I believe that the explanation is unambiguous regarding the use of concepts and relations.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

4. **Parsimony** *

I believe that the explanation is presented in a way that is not too complex.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

5. **Completeness** *

I believe that the explanation provides sufficient information to explain a new drug candidate.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

6. **Soundness** *

I believe that the information paths shown in the explanation are useful for finding potential drug candidates.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

7. Do you have any specific comments about this explanation? If so, please enter these comments below:

#2 Explainable AI for DMD Drug-Repurposing

Obtaining drug candidates that can treat symptoms observed in Duchenne muscular dystrophy disease

* Indicates required question

This questionnaire is part of a Leiden University master's thesis project performed by Rosa Zwart in collaboration with the LUMC BioSemantics group. This project aims to investigate whether the explainability of explainable AI algorithms can be improved by using Foundational Ontologies.

Introduction

This questionnaire is meant for obtaining feedback from users with knowledge and experience in drug repurposing and/or the rare disease DMD.

A knowledge graph is created containing phenotypic and genetic information directly and indirectly associated with Duchenne muscular dystrophy (DMD). Subsequently, data have been incorporated into the graph that relate to drugs and their associated protein targets as well as information regarding the symptoms and diseases for which a drug is used during treatment. Using this knowledge graph, a deep learning model is trained which results in predictions of new links between drugs and phenotypes directly associated with DMD. Due to the black-box nature of deep learning algorithms, it is difficult for human decision-makers to assess the reliability of these predictions. To enable decision-makers to subject the predictions obtained by the model to their own reasoning, interpretable explanations are added to the link predictions of the trained graph neural network (GNN) model. By adding an explanation-generating method to the deep learning algorithm, we arrive at the field of explainable AI that already shows promising developments into the direction of applying trustworthy deep learning techniques to important medical decisions.

During this research project, we want to investigate whether the explanations generated for the predictions by the trained model can be improved. To attempt this, we tested two different approaches, and we are now running a blind assessment of the results. In this questionnaire, you got an explanation generated by one of the two approaches.

The explanations are subgraphs that explain the predicted existence of a certain link between a drug and DMD-related symptom node by showing the most impactful links that lead to this prediction.

Background Information

In order to acquire an idea about the background of all participants, some questions will be asked regarding your research experience.

1. How many years of experience do you have in the field of drug repurposing and/or rare diseases research? *

   ____

2. Do you conduct research on the rare disease Duchenne muscular dystrophy? *

   Mark only one oval.

   ○ Yes

   ○ No

Explanation

To assess the explainability of the given explanation, Likert scaled scores need to be assigned for each explainability aspect.

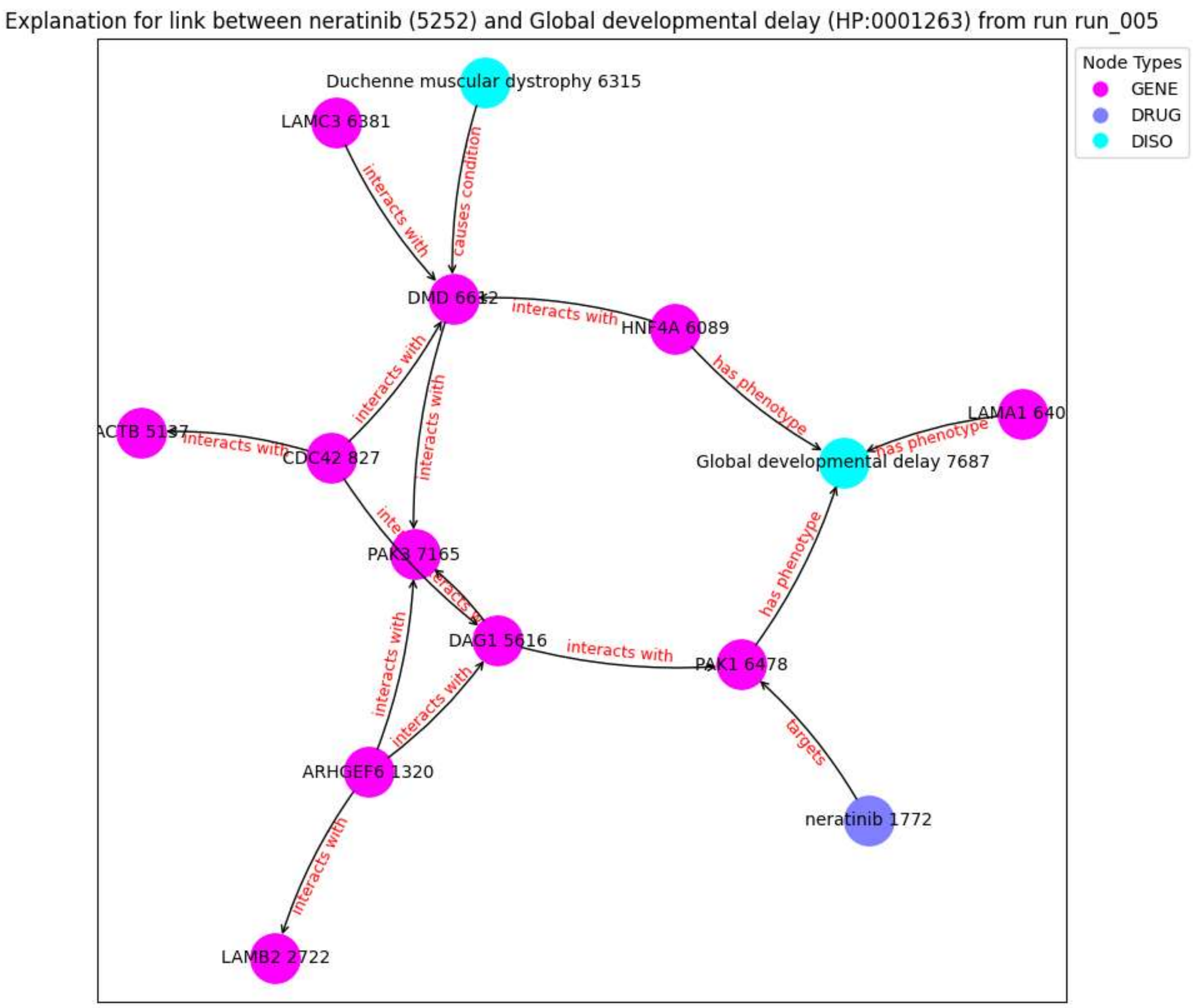
A caption is added to the subgraph to show the meaning of the most relevant paths for explaining the prediction and to provide additional information about the instances found in the graphs.

Prediction

It has been predicted that neratinib treats global developmental delay.

Explanation (Check the graph before reading the text below)

Neratinib targets the gene PAK1. PAK1 interacts with DAG1. DAG1 interacts with PAK3 and CDC42. Both CDC42 and PAK3 interact with DMD. DMD interact with HNF4A. Mutations in HNF4A and PAK1 can cause the phenotype global developmental delay.


Explanation for link between neratinib (5252) and Global developmental delay (HP:0001263) from run run_005

3. **Clarity** *

I believe that the explanation is unambiguous regarding the use of concepts and relations.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

4. **Parsimony** *

I believe that the explanation is presented in a way that is not too complex.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

5. **Completeness** *

I believe that the explanation provides sufficient information to explain a new drug candidate.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

6. **Soundness** *

I believe that the information paths shown in the explanation are useful for finding potential drug candidates.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

7. Do you have any specific comments about this explanation? If so, please enter these comments below:

#4 Explainable AI for DMD Drug-Repurposing

Obtaining drug candidates that can treat symptoms observed in Duchenne muscular dystrophy disease

* Indicates required question

This questionnaire is part of a Leiden University master's thesis project performed by Rosa Zwart in collaboration with the LUMC BioSemantics group. This project aims to investigate whether the explainability of explainable AI algorithms can be improved by using Foundational Ontologies.

Introduction

This questionnaire is meant for obtaining feedback from users with knowledge and experience in drug repurposing and/or the rare disease DMD.

A knowledge graph is created containing phenotypic and genetic information directly and indirectly associated with Duchenne muscular dystrophy (DMD). Subsequently, data have been incorporated into the graph that relate to drugs and their associated protein targets as well as information regarding the symptoms and diseases for which a drug is used during treatment. Using this knowledge graph, a deep learning model is trained which results in predictions of new links between drugs and phenotypes directly associated with DMD. Due to the black-box nature of deep learning algorithms, it is difficult for human decision-makers to assess the reliability of these predictions. To enable decision-makers to subject the predictions obtained by the model to their own reasoning, interpretable explanations are added to the link predictions of the trained graph neural network (GNN) model. By adding an explanation-generating method to the deep learning algorithm, we arrive at the field of explainable AI that already shows promising developments into the direction of applying trustworthy deep learning techniques to important medical decisions.

During this research project, we want to investigate whether the explanations generated for the predictions by the trained model can be improved. To attempt this, we tested two different approaches, and we are now running a blind assessment of the results. In this questionnaire, you got an explanation generated by one of the two approaches.

The explanations are subgraphs that explain the predicted existence of a certain link between a drug and DMD-related symptom node by showing the most impactful links that lead to this prediction.

Background Information

In order to acquire an idea about the background of all participants, some questions will be asked regarding your research experience.

1. How many years of experience do you have in the field of drug repurposing and/or rare diseases research? *

   ____

2. Do you conduct research on the rare disease Duchenne muscular dystrophy? *

   Mark only one oval.

   ○ Yes

   ○ No

Explanation

To assess the explainability of the given explanation, Likert scaled scores need to be assigned for each explainability aspect.

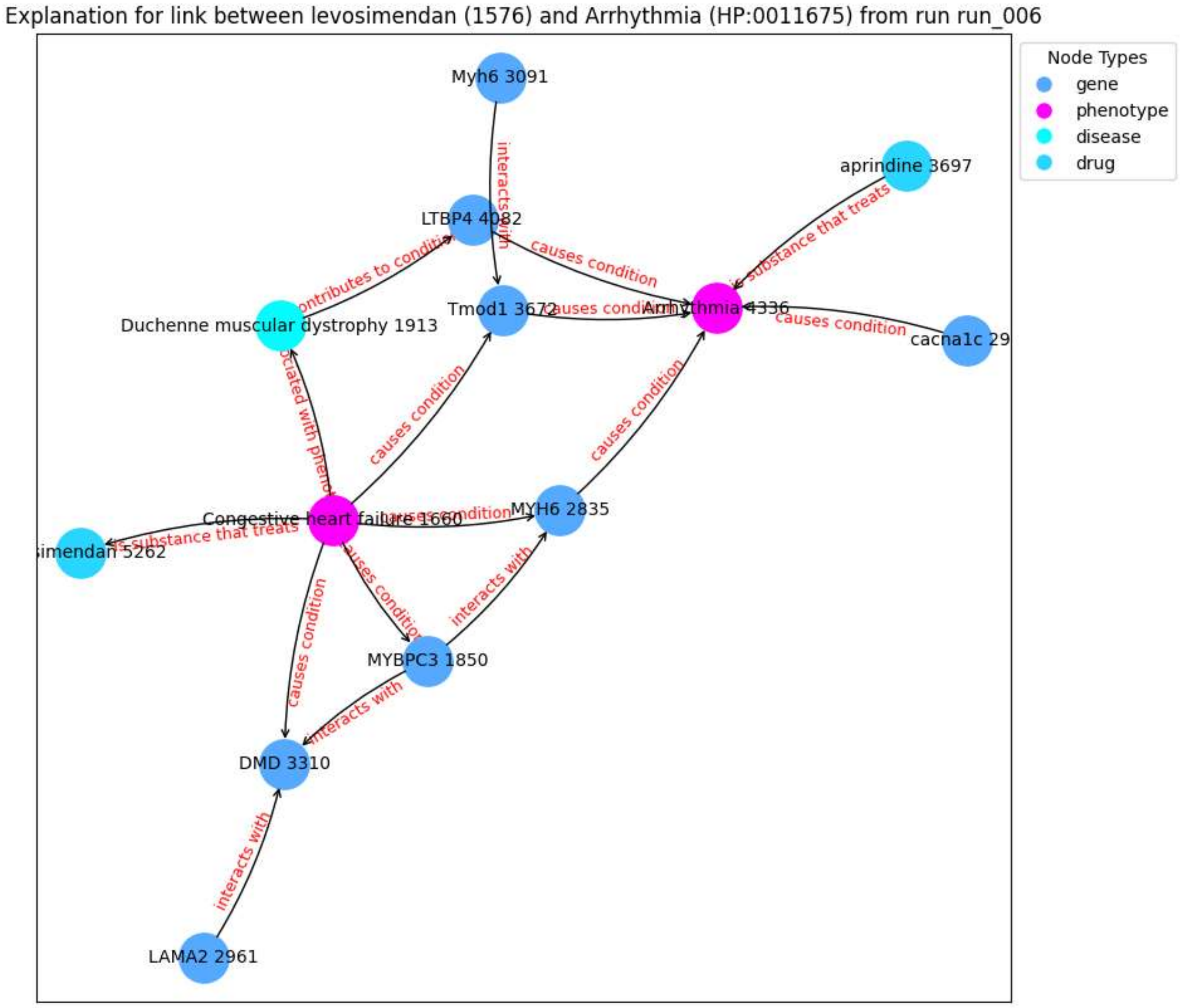
A caption is added to the subgraph to show the meaning of the most relevant paths for explaining the prediction and to provide additional information about the instances found in the graphs.

Prediction

It has been predicted that levosimendan treats arrhythmia.

Explanation (Check the graph before reading the text below)

Levosimendan is a substance that treats congestive heart failure. A mutation in genes DMD, MYBPC3, MYH6, Tmod1 can cause the condition congestive heart failure. MYBPC3 interacts with DMD and MYH6. Changes in genes MYH6 and Tmod1 can cause the condition arrhythmia.



3. **Clarity** *

I believe that the explanation is unambiguous regarding the use of concepts and relations.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

4. **Parsimony** *

I believe that the explanation is presented in a way that is not too complex.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

5. **Completeness** *

I believe that the explanation provides sufficient information to explain a new drug candidate.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

6. **Soundness** *

I believe that the information paths shown in the explanation are useful for finding potential drug candidates.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

7. Do you have any specific comments about this explanation? If so, please enter these comments below:

#5 Explainable AI for DMD Drug-Repurposing

Obtaining drug candidates that can treat symptoms observed in Duchenne muscular dystrophy disease

* Indicates required question

This questionnaire is part of a Leiden University master's thesis project performed by Rosa Zwart in collaboration with the LUMC BioSemantics group. This project aims to investigate whether the explainability of explainable AI algorithms can be improved by using Foundational Ontologies.

Introduction

This questionnaire is meant for obtaining feedback from users with knowledge and experience in drug repurposing and/or the rare disease DMD.

A knowledge graph is created containing phenotypic and genetic information directly and indirectly associated with Duchenne muscular dystrophy (DMD). Subsequently, data have been incorporated into the graph that relate to drugs and their associated protein targets as well as information regarding the symptoms and diseases for which a drug is used during treatment. Using this knowledge graph, a deep learning model is trained which results in predictions of new links between drugs and phenotypes directly associated with DMD. Due to the black-box nature of deep learning algorithms, it is difficult for human decision-makers to assess the reliability of these predictions. To enable decision-makers to subject the predictions obtained by the model to their own reasoning, interpretable explanations are added to the link predictions of the trained graph neural network (GNN) model. By adding an explanation-generating method to the deep learning algorithm, we arrive at the field of explainable AI that already shows promising developments into the direction of applying trustworthy deep learning techniques to important medical decisions.

During this research project, we want to investigate whether the explanations generated for the predictions by the trained model can be improved. To attempt this, we tested two different approaches, and we are now running a blind assessment of the results. In this questionnaire, you got an explanation generated by one of the two approaches.

The explanations are subgraphs that explain the predicted existence of a certain link between a drug and DMD-related symptom node by showing the most impactful links that lead to this prediction.

Background Information

In order to acquire an idea about the background of all participants, some questions will be asked regarding your research experience.

1. How many years of experience do you have in the field of drug repurposing and/or rare diseases research? *

   ____

2. Do you conduct research on the rare disease Duchenne muscular dystrophy? *

   Mark only one oval.

   ○ Yes

   ○ No

Explanation

To assess the explainability of the given explanation, Likert scaled scores need to be assigned for each explainability aspect.

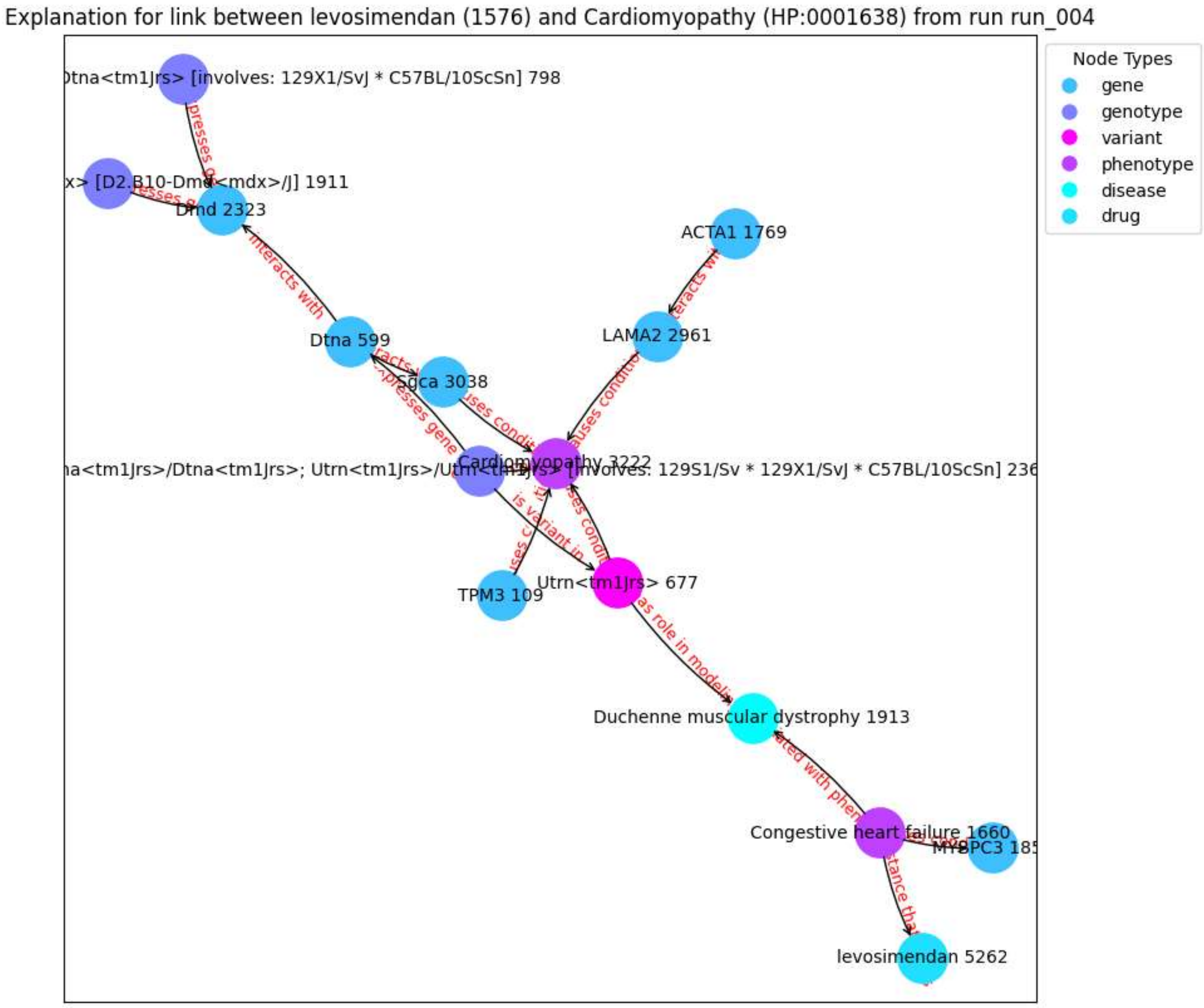
A caption is added to the subgraph to show the meaning of the most relevant paths for explaining the prediction and to provide additional information about the instances found in the graphs.

Prediction

It has been predicted that levosimendan treats cardiomyopathy.

Explanation (Check the graph before reading the text below)

Levosimendan is a substance that treats congestive heart failure. This phenotype is associated with Duchenne muscular dystrophy. The variant Utrn<tm1Jrs> with identifier MGI:2148539 has a role in modeling Duchenne muscular dystrophy. Utrn<tm1Jrs> is a variant found in genotype Dmd<mdx>; Dtna<tm1Jrs>/Dtna<tm1Jrs>; Utrn<tm1Jrs>/Utrn<tm1Jrs> [involves: 129S1/Sv * 129X1/SvJ * C57BL/10ScSn] with identifier MGI:2176891. In this genotype the gene Dtna is expressed. This gene interacts with Sgca. Changes in Sgca can cause the condition cardiomyopathy as well as the previously mentioned variant Utrn<tm1Jrs>.


Explanation for link between levosimendan (1576) and Cardiomyopathy (HP:0001638) from run run_004

3. **Clarity** *

I believe that the explanation is unambiguous regarding the use of concepts and relations.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

4. **Parsimony** *

I believe that the explanation is presented in a way that is not too complex.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

5. **Completeness** *

I believe that the explanation provides sufficient information to explain a new drug candidate.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

6. **Soundness** *

I believe that the information paths shown in the explanation are useful for finding potential drug candidates.

Mark only one oval.

	1	2	3	4	5	
Stro	○	○	○	○	○	Strongly applies

7. Do you have any specific comments about this explanation? If so, please enter these comments below:

____

____

____

____

____

E.2 Responses

Table E.1: All responses for each rating question about the aspects of the explainability of the explanation per questionnaire.

Model Input	Questionnaire	How many years of experience do you have in the field of drug repurposing and/or rare diseases research?	Do you conduct research on the rare disease Duchenne muscular dystrophy?	(Clarity) I believe that the explanation is unambiguous regarding the use of concepts and relations.	(Parsimony) I believe that the explanation is presented in a way that is not too complex.	(Completeness) I believe that the explanation provides sufficient information to explain a new drug candidate.	(Soundness) I believe that the information paths shown in the explanation are useful for finding potential drug candidates.
Original Knowledge Graph	#1 Figure D.9	19	Yes	4	4	2	4
		25	Yes	2	2	3	3
	#2 Figure D.3	23	Yes	2	1	2	4
		3	Yes	2	5	3	4
		2	Yes	4	5	3	4
Restructured Knowledge Graph	#4 Figure D.25	3	No	3	4	3	5
		0.5	Yes	2	2	3	4
		9	Yes	2	4	3	2
	#5 Figure D.14	10	No	1	1	1	1
		0.5	Yes	4	2	4	2

Table E.2: Additional comments given by the respondents about the shown explanation for each questionnaire.

Model Input	Questionnaire	Do you have any specific comments about this explanation?
Original Knowledge Graph	#1 Figure D.9	The drug targets pak1, however the context in which the interaction of pak1 and binders related to the phenotype is not specified. Does this happen in cell types and brain areas related to language? pak1 has functions in different cell types. I am no pak1 expert, but a quick google search shows that pak1 mutations have also been linked to speech delay and intellectual disability, which are relevant for DMD. I also miss the way the drug targets pak1. Will it inhibit or stimulate the target? DMD is a pediatric disease; has the drug been used/tested and deemed safe in pediatric populations? The drug also gives muscle spasms, which do not seem a workable side effect to have in children where the primary pathology is in muscle tissue.
	#2 Figure D.3	Sorry but the text is too complex with many gene names and interactions and then the figure is not very clear either because there seems to be extra information in there that is not in the text? It was difficult to match the graph with the text.
		To understand the figure it is not needed to understand the blue 'DISO' dot in the legend, but it did trigger me to think "Hmm what does that mean, why is it labeled? Do I need to understand"? Maybe if it is not relevant information leave it out. Or if it is relevant information label it with a term that is more generally understood.
Restructured Knowledge Graph	#4 Figure D.25	I don't have any experience with these kind of graphs so I'm not sure whether it's because of my lack of knowledge that I find the graph a bit unclear
		Some genes are in there multiple times (e.g. MYH6, but also murine Myh6?). Also some arrows seem to be flipped around (a disease causing a gene). DMD and Duchenne muscular dystrophy are not linked, which is odd.
	#5 Figure D.14	I did not at all understand what you are trying to say and explain. I tried to read it multiple times but the way the information and data is presented is absolutely not clear.
		I'm not sure if it's my lack of knowledge or whether the graph was just not so clear to interpret. Besides that, the texts (especially the red ones) were not readable well

Bibliography

[1] Deepsnap 0.2.0 documentation. `https://snap.stanford.edu/deepsnap/`.

[2] National center for biotechnology information (ncbi). [Cited 2023 Nov 22] https://www.ncbi.nlm.nih.gov/.

[3] Pytorch 1.11.0 documentation. `https://pytorch.org/docs/1.11/`.

[4] Pytorch 1.11.0 documentation on *BCEWithLogitsLoss*. `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html`.

[5] *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*. Springer International Publishing, 2022.

[6] HLEG AI. High-level expert group on artificial intelligence, 2019.

[7] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS ONE*, 9(1):e85777, jan 2014.

[8] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. *CoRR*, abs/1806.07538, 2018.

[9] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[10] César Henrique Bernabé, Núria Queralt-Rosinach, Vítor Silva Souza, Luiz Olavo Bonino da Silva Santos, Barend Mons, Annika Jacobsen, and Marco Roos. The use of foundational ontologies in bioinformatics, 08 2022.

[11] Stefano Borgo, Roberta Ferrario, Aldo Gangemi, Nicola Guarino, Claudio Masolo, Daniele Porello, Emilio M. Sanfilippo, and Laure Vieu. Dolce: A descriptive ontology for linguistic and cognitive engineering1. *Applied Ontology*, 17(1):45–69, March 2022.

[12] Matthew Brush, Nico Matentzoglu, and Melissa Haendel. Geno ontology, 2023.

[13] The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 11 2022.

[14] Tukur Dahiru. P - value, a true test of statistical significance? a cautionary note. *Ann. Ib. Postgrad. Med.*, 6(1):21–26, June 2008.

[15] Fabiano Dalpiaz, Xavier Franch, and Jennifer Horkoff. istar 2.0 language guide. *CoRR*, abs/1605.07767, 2016.

[16] Jeffrey Dean. A golden decade of deep learning: Computing systems and applications. *Daedalus*, 151(2):58–74, 05 2022.

[17] 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, page 135–144, New York, NY, USA, 2017. Association for Computing Machinery.

[18] Bruno Borlini Duarte. *An Ontology-based Reference Model for the Software Systems Domain with a focus on Requirements Traceability.* PhD thesis, Universidade Federal do Espírito Santo – UFES, 2022.

[19] Mustafa Berk Duran and Gunter Mussbacher. Reusability in goal modeling: A systematic literature review. *Information and Software Technology*, 110:156–173, 2019.

[20] Ivo F A C Fokkema, Peter E M Taschner, Gerard C P Schaafsma, J Celli, Jeroen F J Laros, and Johan T den Dunnen. LOVD v.2.0: the next generation in gene variant databases. *Hum. Mutat.*, 32(5):557–563, May 2011.

[21] Zheng Gao, Gang Fu, Chunping Ouyang, Satoshi Tsutsui, Xiaozhong Liu, Jeremy Yang, Christopher Gessner, Brian Foote, David Wild, Ying Ding, and Qi Yu. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. *BMC Bioinformatics*, 20(1):306, Jun 2019.

[22] Georgios Gkoutos, Paul Schofield, and Robert Hoehndorf. The anatomy of phenotype ontologies: principles, properties and applications. *Briefings in bioinformatics*, 19, 04 2017.

[23] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[24] Nicola Guarino. Formal ontologies and information systems. 06 1998.

[25] Giancarlo Guizzardi. *Ontological foundations for structural conceptual models.* PhD thesis, University of Twente, October 2005.

[26] Giancarlo Guizzardi. On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In *International Baltic Conference on Databases and Information Systems*, 2007.

[27] Giancarlo Guizzardi. Ontology, Ontologies and the "I" of FAIR. *Data Intelligence*, 2(1-2):181–191, 01 2020.

[28] Giancarlo Guizzardi, Alessander Bottes Benevides, Claudemir M. Fonseca, Jo ao Paulo A. Almeida, Tiago Prince Sales, and Daniele Porello. Ufo: Unified foundational ontology. *Applied ontology*, 1(17):167–210, 2022.

[29] Giancarlo Guizzardi, Anna Bernasconi, Oscar Pastor, and Veda Storey. *Ontological Unpacking as Explanation: The Case of the Viral Conceptual Model*, pages 356–366. 10 2021.

[30] Melissa A. Haendel, Fabian Neuhaus, David Osumi-Sutherland, Paula M. Mabee, Jos L.V. Mejino, Chris J. Mungall, and Barry Smith. *CARO – The Common Anatomy Reference Ontology*, pages 327–349. Springer London, London, 2008.

[31] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Ga editor, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[32] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[33] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, jul 2021.

[34] Andreas Holzinger, Randy Goebel, Ruth Fong, Taesup Moon, Klaus-Robert Müller, and Wojciech Samek. *xxAI - Beyond Explainable Artificial Intelligence*, pages 3–10. Springer International Publishing, Cham, 2022.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[36] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLoS One*, 9(6):e98679, June 2014.

[37] Kavisha Jayasundara, Aidan Hollis, Murray Krahn, Muhammad Mamdani, Jeffrey S. Hoch, and Paul Grootendorst. Estimating the clinical

cost of drug development for orphan versus non-orphan drugs. *Orphanet Journal of Rare Diseases*, 14(1), jan 2019.

[38] Simon Jupp, Tony Burdett, Catherine Leroy, and Helen E. Parkinson. A new ontology lookup service at embl-ebi. In *Workshop on Semantic Web Applications and Tools for Life Sciences*, 2015.

[39] C.M. Keet. *An Introduction to Ontology Engineering.* College Publications, 2020.

[40] Sebastian Köhler, Michael Gargano, Nicolas Matentzoglu, Leigh C Carmody, David Lewis-Smith, Nicole A Vasilevsky, Daniel Danis, Ganna Balagura, Gareth Baynam, Amy M Brower, Tiffany J Callahan, Christopher G Chute, Johanna L Est, Peter D Galer, Shiva Ganesan, Matthias Griese, Matthias Haimel, Julia Pazmandi, Marc Hanauer, Nomi L Harris, Michael J Hartnett, Maximilian Hastreiter, Fabian Hauck, Yongqun He, Tim Jeske, Hugh Kearney, Gerhard Kindle, Christoph Klein, Katrin Knoflach, Roland Krause, David Lagorce, Julie A McMurry, Jillian A Miller, Monica C Munoz-Torres, Rebecca L Peters, Christina K Rapp, Ana M Rath, Shahmir A Rind, Avi Z Rosenberg, Michael M Segal, Markus G Seidel, Damian Smedley, Tomer Talmy, Yarlalu Thomas, Samuel A Wiafe, Julie Xian, Zafer Yüksel, Ingo Helbig, Christopher J Mungall, Melissa A Haendel, and Peter N Robinson. The human phenotype ontology in 2021. *Nucleic Acids Research*, 49(D1):D1207–D1217, 12 2020.

[41] Tom Lenaerts. *Scale Free Networks*, pages 1492–1493. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[42] Jure Leskovec. Stanford cs224w: A general perspective on gnns.

[43] Jure Leskovec. Stanford cs224w: Training graph neural networks.

[44] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[45] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[46] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021.

[47] Frank Loebe, Patryk Burek, and Heinrich Herre. Gfo: The general formal ontology. *Applied Ontology*, 17(1):71–106, March 2022.

[48] F. J. López-Pellicera, L. M. Vilches-Blázquezb, J. Nogueras-Isoa, O. Corchoc, M. A. Bernabéd, and A. F. Rodríguezb. Using a hybrid approach for the development of an ontology in the hydrographical domain. 2007.

[49] Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, 2021.

[50] Ana Ma Martínez Ferrandis, Oscar Pastor López, and Giancarlo Guizzardi. Applying the principles of an ontology-based approach to a conceptual schema of human genome. In Wilfred Ng, Veda C. Storey, and Juan C. Trujillo, editors, *Conceptual Modeling*, pages 471–478, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[51] David Moreira. *Orthologous Gene*, pages 1–1. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[52] Chris Mungall, Nico Matentzoglu, Jim Balhoff, David Osumi-Sutherland, pgaudet, Shawn Tan, Bill Duncan, Clare Pilgrim, James A. Overton, Charles Tapley Hoyt, Lauren, Nomi Harris, Sierra Moxon, lschriml, Nicole Vasilevsky, sabrinatoro, Matthew Brush, Vasundra Touré, Anita Caron, Damien Goutte-Gattat, Michael Sinclair, Anthony Bretaudeau, Scott Cain, Melissa Haendel, diatomsRcool, Bide Zhang, actions user, Jen Hammock, and Marie-Angélique Laporte. oborel/obo-relations: February 2023 release 2 (major pipeline change), February 2023.

[53] Francis Sahngun Nahm. Receiver operating characteristic curve: overview and practical use for clinicians. *Korean J. Anesthesiol.*, 75(1):25–36, February 2022.

[54] Chao Pang, Annet Sollie, Anna Sijtsma, Dennis Hendriksen, Bart Charbon, Mark de Haan, Tommy de Boer, Fleur Kelpin, Jonathan Jetten, Joeri K van der Velde, Nynke Smidt, Rolf Sijmons, Hans Hillege, and Morris A Swertz. SORTA: a system for ontology-based re-coding and technical annotation of biomedical phenotype data. *Database (Oxford)*, 2015:bav089, September 2015.

[55] Bastian Pfeifer, Anna Saranti, and Andreas Holzinger. Gnn-subnet: disease subnetwork detection with explainable graph neural networks. *Bioinformatics*, 38(Supplement2):ii120–ii126, September 2022.

[56] Núria Queralt-Rosinach, Gregory S Stupp, Tong Shu Li, Michael Mayers, Maureen E Hoatlin, Matthew Might, Benjamin M Good, and Andrew I Su. Structured reviews for data and knowledge-driven research. *Database (Oxford)*, 2020, January 2020.

[57] Pablo Perdomo Quinteiro. Dmd drug repurposing using knowledge graphs and explainable ai. Master's thesis, Leiden University, 2022.

[58] Ruth L Seal, Bryony Braschi, Kristian Gray, Tamsin E M Jones, Susan Tweedie, Liora Haim-Vilmovsky, and Elspeth A Bruford. Genenames.Org:

The HGNC resources in 2023. *Nucleic Acids Res.*, 51(D1):D1003–D1009, January 2023.

[59] Kent A Shefchek, Nomi L Harris, Michael Gargano, Nicolas Matentzoglu, Deepak Unni, Matthew Brush, Daniel Keith, Tom Conlin, Nicole Vasilevsky, Xingmin Aaron Zhang, James P Balhoff, Larry Babb, Susan M Bello, Hannah Blau, Yvonne Bradford, Seth Carbon, Leigh Carmody, Lauren E Chan, Valentina Cipriani, Alayne Cuzick, Maria Della Rocca, Nathan Dunn, Shahim Essaid, Petra Fey, Chris Grove, Jean-Phillipe Gourdine, Ada Hamosh, Midori Harris, Ingo Helbig, Maureen Hoatlin, Marcin Joachimiak, Simon Jupp, Kenneth B Lett, Suzanna E Lewis, Craig McNamara, Zoë M Pendlington, Clare Pilgrim, Tim Putman, Vida Ravanmehr, Justin Reese, Erin Riggs, Sofia Robb, Paola Roncaglia, James Seager, Erik Segerdell, Morgan Similuk, Andrea L Storm, Courtney Thaxon, Anne Thessen, Julius O B Jacobsen, Julie A McMurry, Tudor Groza, Sebastian Köhler, Damian Smedley, Peter N Robinson, Christopher J Mungall, Melissa A Haendel, Monica C Munoz-Torres, and David Osumi-Sutherland. The monarch initiative in 2019: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res*, 48(D1):D704–D715, January 2020.

[60] Andrew D. Spear, Werner Ceusters, and Barry Smith. Functions in basic formal ontology. *Applied ontology*, 11(2):103–128, 2016.

[61] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1):161–197, 1998.

[62] Oleg Ursu, Jayme Holmes, Jeffrey Knockel, Cristian G Bologa, Jeremy J Yang, Stephen L Mathias, Stuart J Nelson, and Tudor I Oprea. DrugCentral: online drug compendium. *Nucleic Acids Res.*, 45(D1):D932–D939, January 2017.

[63] Nicole Vasilevsky, Nicolas Matentzoglu, Sabrina Toro, Joseph Flack, Harshad Hegde, Deepak Unni, Gioconda Alyea, Joanna Amberger, Larry Babb, James Balhoff, Taylor Bingaman, Gully Burns, Orion Buske, Tiffany Callahan, Leigh Carmody, Paula Carrio Cordo, Lauren Chan, George Chang, Sean Christiaens, Michel Dumontier, Laura Failla, May Flowers, Alpha Garrett, Jennifer Goldstein, Dylan Gration, Tudor Groza, Marc Hanauer, Nomi Harris, Jason Hilton, Daniel Himmelstein, Charles Tapley Hoyt, Megan Kane, Sebastian Köhler, David Lagorce, Abbe Lai, Martin Larralde, Antonia Lock, Irene López Santiago, Donna Maglott, Adriana Malheiro, Birgit Meldal, Monica Munoz-Torres, Tristan Nelson, Frank Nicholas, David Ochoa, Daniel Olson, Tudor Oprea, David Osumi-Sutherland, Helen Parkinson, Zoë May Pendlington, Ana Rath, Heidi Rehm, Lyubov Remennik, Erin Riggs, Paola Roncaglia, Justyne Ross, Marion Shadbolt, Kent Shefchek, Morgan Similuk, Nicholas

Sioutos, Damian Smedley, Rachel Sparks, Ray Stefancsik, Ralf Stephan, Andrea Storm, Doron Stupp, Gregory Stupp, Jagadish Chandrabose Sundaramurthi, Imke Tammen, Darin Tay, Courtney Thaxton, Eloise Valasek, Jordi Valls-Margarit, Alex Wagner, Danielle Welter, Patricia Whetzel, Lori Whiteman, Valerie Wood, Colleen Xu, Andreas Zankl, Xingmin Aaron Zhang, Christopher Chute, Peter Robinson, Christopher Mungall, Ada Hamosh, and Melissa Haendel. Mondo: Unifying diseases for the world, by the world, 2022.

[64] V Venugopal and S Pavlakis. Duchenne muscular dystrophy. [Updated 2023 Jul 10] `https://www.ncbi.nlm.nih.gov/books/NBK482346/`.

[65] Raoul R. Wadhwa and Raghavendra Marappa-Ganeshan. T test, 2023. [Updated 2023 Jan 16] Available from: `https://www.ncbi.nlm.nih.gov/books/NBK553048/`.

[66] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1), March 2016.

[67] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks, 2019.

[68] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[69] Ying Zhou, Yintao Zhang, Donghai Zhao, Xinyuan Yu, Xinyi Shen, Yuan Zhou, Shanshan Wang, Yunqing Qiu, Yuzong Chen, and Feng Zhu. TTD: Therapeutic target database describing target druggability information. *Nucleic Acids Res.*, September 2023.