Bachelor thesis

Artificial Intelligence

Radboud University 

Graph Similarity classification on HPO graphs of patients with a genetic disorder

Author:
Niels Teunissen
s1020104

First supervisor:
drs. Lex Dingemans
Radboudumc
A.Dingemans@radboudumc.nl

Second supervisor:
dr. Max Hinne
Artificial Intelligence
m.hinne@donders.ru.nl

January 25, 2022

Abstract

1-3% of the general population has intellectual disability. While many of these patients can not be diagnosed with a gene deficit, in 10-20% of these patients, only a variant of unknown significance (VUS) is found. A variant of unknown significance is a genetic variant that has been identified, but whose significance to the function or health of the person is unknown. A VUS makes diagnosis impossible.

Therefore this study aims to find a method for diagnosing patients with a VUS. We predict the gene deficit by using graph similarity classification on human phenotype ontology (HPO) graphs of patients with a genetic disorder. The human phenotype ontology is a standardized vocabulary of phenotypic abnormalities encountered in human disease [9]. To achieve our aim we ask the following research question: Can we accurately classify which genetic disorder a patient has based on HPO graph similarity classification?

We have implemented 2 graph similarity classification methods to classify patients with a VUS. First of all, most common subgraph classification based on one of the following metrics: amount of nodes, amount of edges, sum of weights, and the amount of nodes with penalty in the most common subgraph. Secondly, we applied graph kernels from GraKeL [10] in combination with support vector machines for classification.

The most important result from the graph similarity classification methods is as follows. The best performing graph kernel for kernel-based graph classification is the vertex histogram kernel with an accuracy of 80% and a F1-score of 0.75. Furthermore, the optimal most common subgraph (MCS) model is the MCS model based on the nodes with penalty metric with an accuracy of 74% and a F1-score of 0.68.

Furthermore, the following 3 conclusions can be drawn from this study. To begin, we can accurately classify which genetic disorder a patient has based on HPO graph similarity classification. Moreover, the simple vertex histogram kernel performs on par with more sophisticated graph kernels such as the Weisfeiler-Lehman optimal assignment kernel, this is likely due to the structure of the HPO graphs. In addition, this model could be used as a tool for clinical geneticists to predict the gene deficit given that the suspected gene deficit is in the data.

Despite these promising results, there are 3 limitations that can be improved in future work. First of all, the data is limited, therefore we suggest adding

more graphs of patients with different gene deficits to the data. Secondly, we suggest adding more useful information to the node and/or edge labels to improve classification with sophisticated kernels. Finally, one can add informative feature vectors to the nodes to make classification with graph neural networks possible.

Contents

Chapter 1

Introduction

1-3% of the general population has intellectual disability. Intellectual disability is a term used to define patients that have a neurodevelopmental disorder in which they have significantly impaired intellectual functioning and an IQ of at most 70. Patients with severe intellectual disability are usually advised by their general practitioner to visit a clinical geneticist for further examination. The geneticist performs whole exome sequencing on the patient and assesses the gene variants; Whole exome sequencing (WES) is a technique for sequencing all of the protein-coding regions of genes in the genome of a human. After sequencing, a gene variant can be benign (not malignant), pathogenic, or a variant of unknown significance (VUS). A variant of unknown significance is a genetic variant that has been identified through WES, but whose significance to the function or health of the person is unknown. If a pathogenic gene variant is found, the clinical geneticist presents a diagnosis to the patient. However, 10-20% of the patients only have a VUS and therefore diagnosis is impossible.

Not having a diagnosis has several disadvantages for the patient and their loved ones. To begin, the effect of the gene mutation is unknown. In addition, prime treatment is harder to facilitate without diagnosis. Furthermore, there is no closure for the patient and family.

To illustrate how this might work in practice, we discuss the hypothetical case of a patient named Paul. Paul is 8 years old and has a range of phenotypic abnormalities such as abnormal behavior, intellectual disability, and an abnormality of the forehead. A phenotypic abnormality is an abnormality that a clinical geneticist diagnosis a patient with. This can include a facial abnormality or an abnormal heart rhythm. His general practitioner refers Paul to consult a clinical geneticist. The geneticist suspects that Paul has an abnormal gene expression. Consequently, Paul undergoes whole exome sequencing, however no pathogenic gene variant is found. Only a variant

of unknown significance. Therefore the clinical geneticist can not diagnose Paul. As a consequence, Paul's parents do not know how the symptoms will progress, treatment can not be improved, and Paul and his family have no closure.

The aim of the study is to diagnose patients with a VUS. We want to solve the variant of unknown significance (VUS) problem based on human phenotype ontology (HPO) graph similarity classification. The human phenotype ontology is a standardized vocabulary of phenotypic abnormalities encountered in human disease [9]. To achieve our aim we ask the following research question: Can we accurately classify which genetic disorder a patient has based on HPO graph similarity classification?

In order to classify patients without a genetic disorder, we need to imitate patients with a VUS. This is because we only have data of patients with a genetic disorder label. Therefore, we simulate that patients have a VUS by selectively removing the label.

We have implemented 2 models in which we can classify patients with a VUS based on HPO graph similarity classification. In our first model, we compare patients that have a VUS with the data of patients that have one of 16 genetic disorders by creating the most common subgraph between this patient and all other patients. Subsequently, we classify the patient that has the VUS with the label of the patient in the largest most common subgraph. A larger most common subgraph implicates that 2 patients have more shared pairs of phenotypic abnormalities, thus being more similar. Therefore, being more likely to have the same genetic disorder. With this model, we can classify which genetic deficit a patient with a VUS probably has. In our second model, we use kernel-based graph classification. We employ 4 different graph kernels in combination with support vector machines to classify patients without a label.

Most common subgraph classification and kernel-based graph classification have been used in other domains such as chemoinformatics, bioinformatics and neuroscience [1], [2], [3], [4], [11]. This offers reason to explore these classification approaches within other domains. Our research contributes to the existing literature by exploring these classification approaches within the domain of clinical genetics.

Chapter 2

Preliminaries

2.0.1 Graphs

There are 4 important properties of our graphs. To begin, a graph G consists of a finite set of nodes N and edges E where $\mathbf{E} \subseteq \{(\mathbf{u}, \mathbf{v}) \subseteq \mathbf{N} | \mathbf{u} \neq \mathbf{v}\}$. Furthermore, $(\mathbf{N1}, \mathbf{E1})$ is a subgraph of $(\mathbf{N2}, \mathbf{E2})$ if it is a graph where $\mathbf{N1} \subseteq \mathbf{N2}$ and $\mathbf{E1} \subset \mathbf{E2}$. Moreover, a graph can also be an induced subgraph; $(\mathbf{N1}, \mathbf{E1})$ is an induced subgraph of $(\mathbf{N2}, \mathbf{E2})$ if it is a graph where $\mathbf{N1} \subseteq \mathbf{N2}$ and **E1** contains all edges of **E2** which are subsets of **N1**. Consequently, we can combine the definitions of the graph and induced subgraph to explain the most common subgraph. A most common subgraph is a graph that is an induced subgraph of 2 graphs where the amount of nodes are maximized.

Additionally, there are 2 important properties about HPO graphs. All HPO graphs are directed acyclic graphs (DAGs), a DAG is a graph that only has directed edges and contains no cycles. The most specific nodes are the leaf nodes in the DAGs, which is a node that has no incoming edges and atleast one outgoing edge.

2.0.2 Kernel methods

Kernel methods are learning algorithms that learn by comparing points of data using similarity measures called kernels.
What is a graph kernel? It can be defined as follows: A nonempty and finite set of graphs G and $\mathbf{k} : \mathbf{G} \star \mathbf{G} \to \mathbb{R}$ is a function. Then, k is a kernel on G if there is a Hilbert space H and a feature map $\phi : \mathbf{G} \to \mathbf{H}$ such that $\mathbf{k}(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$ for $\mathbf{x}, \mathbf{y} \in \mathbf{G}$ where $\langle ., . \rangle$ denotes the inner product of H. This feature map only exists if k is a positive-semidefinte function [6].

2.0.3 Genetics

There are 5 important definitions about genetics to understand. To begin, intellectual disability is a term used to define patients that have a neurodevelopmental disorder in which they have significantly impaired intellectual functioning (e.g. impaired ability to learn) and adaptive functioning (to cope in your environment with greatest success and least conflict). Moreover, everyone with an IQ of at most 70 has intellectual disability. In addition, patients with intellectual disability usually undergo Whole Exome Sequencing (WES), which is a genomic technique for sequencing all of the protein-coding regions of genes in a genome, which is also known as the exome. The first part of WES is to select the exons which is the DNA that encondes proteins. In the second part, this DNA is sequenced by a DNA-sequencing technique. Resulting in a list of gene variants that are benign, malignant, or are a variant of unknown significance. A variant of unknown significance is a genetic variant that has been identified through WES, but of which the significance to the function or health of the person is unknown. Moreover, it is important to know the distinction between the genotype and phenotype, the genotype is the set of genes of a human. The phenotype is the set of observable characteristics of a human resulting from the interaction between its genotype and the environment.

In addition, there are 3 important definitions about the HPO to understand. First of all, the nodes in the graphs have a HPO term as label and the Human Phenotype Ontology (HPO) is a standardized vocabulary of phenotypic abnormalities encountered in human disease, each term in the HPO describes a phenotypic abnormality, e.g. Lymphangioma (noncancerous, fluid-filled cysts that occur in lymphatic vessels). Secondly, the HPO terms are ordered hierarchically, the further one goes in to the HPO terms the more specific the terms become. For example, one can have a HPO path as follows: All $\rightarrow$ phenotypic abnormality $\rightarrow$ Growth abnormality $\rightarrow$ Acral overgrowth. Each child term is more specific than its parent term. Finally, a path in a HPO graph is a finite set of directed edges, which joins a sequence of distinct vertices.

Chapter 3

Related Work

There have been numerous studies investigating graph similarity classification. First of all, graph similarity classification based on the most common subgraph has been succesfully used in chemoinformatics by Cao *et al.* [2]. They implemented a new backtracking algorithm for the Most Common Subgraph (Maximum Common Substructure) problem, specifically tailored for the chemical structure of graphs. The new MCS method outperformed existing methods that are not specifically made for chemical structures. This highlights the importance of using the MCS in the optimal way for the structure of the graphs. Moreover, graph-kernel based similarity classification is also widely used in other domains such as bioinformatics, neuroscience, natural language processing, and computer vision. For example, Cui *et al.* [3] demonstrates that graph kernels, such as the Weisfeiler-Lehman subtree kernel improves classification and extracts important features from the graphs of patients with Alzheimers's disease, Mild Cognitive Impairment, and normal controls. This study demonstrates that the combination of graph kernels with support vector machines improves classification performance compared to other state-of-the-art methods - hence providing us with an incentive to apply this combination to HPO graphs. Borgwardt *et al.* [1] used a random walk kernel to classify proteins as either enzyme or non-enzyme. They discovered that graph kernels in combination with support vector machines perform as good as vector models when having less protein information in the graphs and outperform vector models when having the same amount of information. Similarly to the previous paper, the combination of graph kernels and support vector machines outperforms existing state-of-the-art methods. Hereto, we expect a better performance when using HPO-graphs instead of using vectors of HPO-terms for patients.

As illustrated, much research has been done in graph similarity classification. However, no research has been done about graph kernels or most common subgraph in combination with the HPO graphs of patients with a genetic

disorder. Nevertheless, other classification methods have been developed with HPO data, such as the Phenomizer by Köhler *et al.* [7]. Phenomizer is a web-based application for differential diagnosis of patients with a diverse set of phenotypic abnormalities, it proposes a list of candidate diseases for a patient based on a significance measure and the Resnik score. Note that they provide a list of candidate diseases and do not classify a patient to one particular disease. Furthermore, Ratnaike *et al.* [8] used similarity scores to distinguish mitochondrial and rare non-mitochondrial disorders patients by comparing patients with a HPO disease profile for mitochondrial disorders. During their research they noted that patient to patient similarity is not an optimal solution due to the overlap in clinical phenotypes between mitochondrial and non-mitochondrial disorders. Thus, even if patient-patient comparisons yield promising results, we do not know if that remains the case if you include different or more genetic disorders.

The following paper by Kriege *et al.* [6] presents an extensive overview of existing graph kernels and reveals that graph kernel classification can be promising in new areas. In order to make this easier, there is also a practitioner's guide in choosing the right graph kernels given the data of the graphs. This paves the way for graph kernel and most common subgraph classification on HPO graphs of patients with a genetic disorder.

Chapter 4

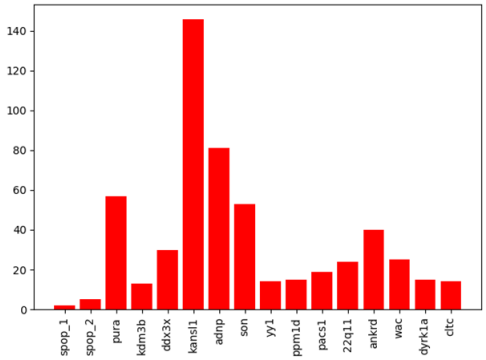
Research

4.0.1 The Data

The data is provided by the Radboudumc and consists of human phenotype ontology data of 553 patients with 16 distinct genetic disorders. The data is a pickle and consists of 553 rows, one for each patient and 6 columns per patient, namely:

- ***hpo_all***: The HPO terms for a patient
- ***hpo_all_name***: The names of the HPO terms for a patient
- ***graphs***: The HPO graph of a patient
- ***hpo_all_with_parents***: The HPO terms of the patient with all hierarchical terms included
- ***hpo_all_name_with_parents***: hpo_all_with_parents with the names of the HPO terms instead of the IDs
- ***label***: the genetic disorder of the patient

Distribution of the genetic disorders



The data is imbalanced with the minority class (*SPOP_1*) having 2 patients and the majority class (*KANSL1*) having 146 patients.

4.0.2 Most Common Subgraph classification

MCS classification is possible on HPO graphs. A MCS is a graph that is an induced subgraph of 2 graphs and has as many vertices as possible. The first model operates based on the most common subgraph (MCS) between patients. However, the MCS problem is NP-complete, which makes exact computation impossible. Nonetheless, the HPO graphs are directed and acyclic, which makes it computationally feasible. Therefore MCS classification is possible on HPO graphs. Let us now take a look at the implementation of the MCS similarity classification.

HPO graph of a patient

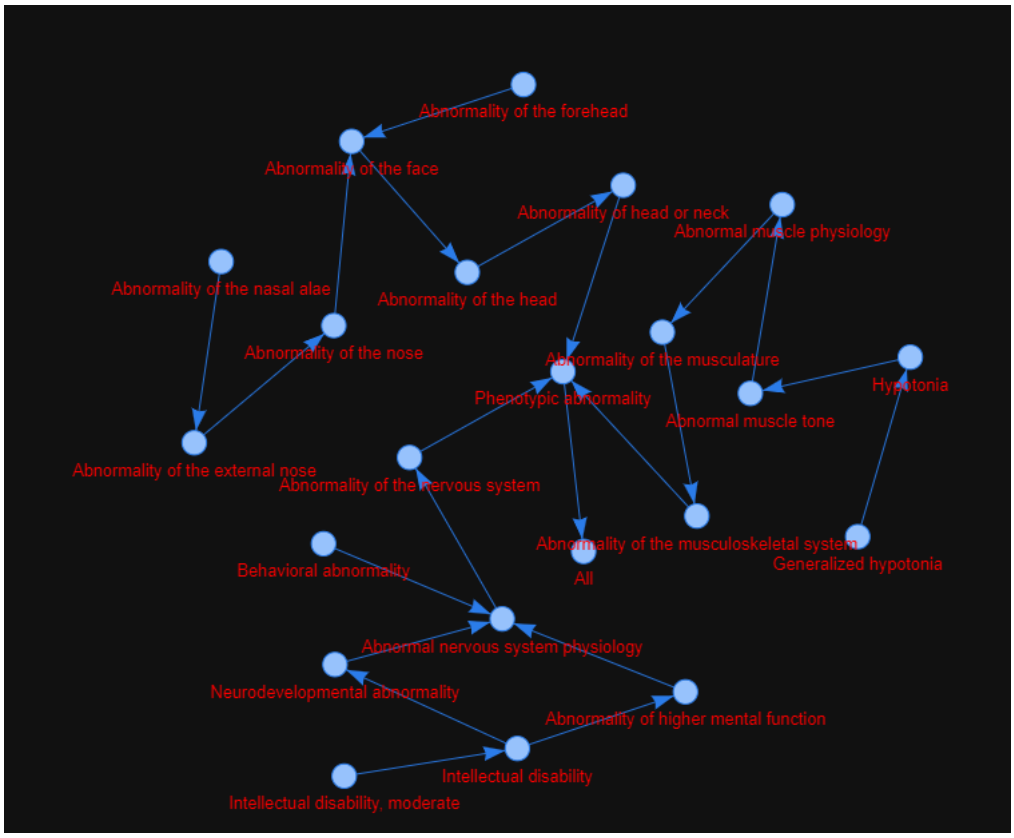


Figure 4.1: Each node in the graph has a HPO term name and the edges are directed from the leaf nodes to the 'All' node. For example, 'Abnormality of the nasal alae' is a more specific phenotypic abnormality than 'Abnormality of the nose' because the length of the path from 'Abnormality of the nasal alae' to the 'All' node is longer.

Calculating the MCS between 2 HPO graphs of patients

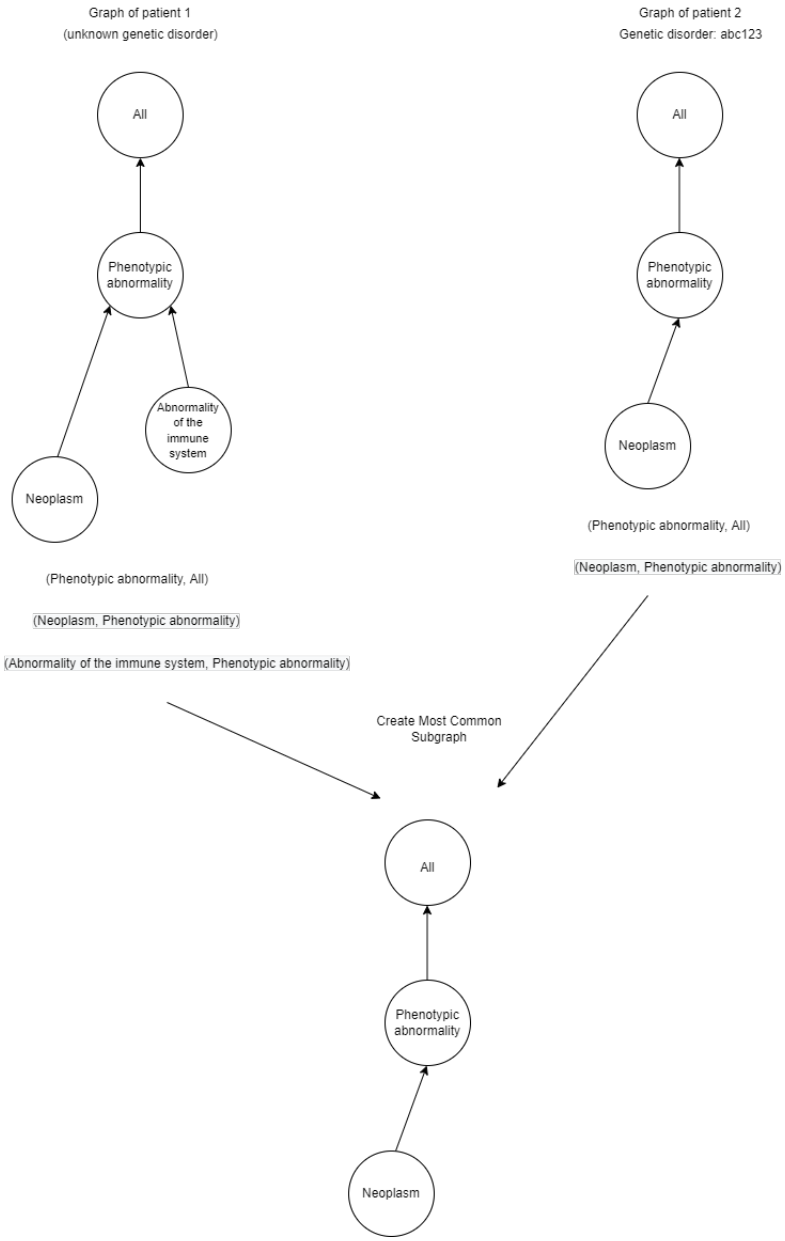


Figure 4.2: The MCS between 2 graphs is calculated by checking if there exists an edge in graph 1 between the node pairs of Graph 2. In this example that is the case for the node pairs (Phenotypic abnormality, All) and (Neoplasm, Phenotypic abnormality).

The algorithm explained:

Most common subgraph classification functions as follows. First of all, we calculate the most common subgraphs between a patient and all other patients, and we repeat this process for all patients - leading to leave-one-out cross-validation (LOO-CV). This process results in 552 most common subgraphs per patient - since the patient is compared to all other patients, but of course not to itself. Moreover, we remove the label from the current patient to imitate a patient with a "VUS", therefore we receive a list of 552 most common subgraphs for each of the 553 'unlabeled' patients. The 'unlabeled' patient is the test set and the other patients are the train set (the same procedure as leave-one-out cross validation). For each unlabeled patient, we look for the most similar most common subgraph in the list of 552 most common subgraphs. The most similar most common subgraph is based on a metric. For example, the number of nodes in the most common subgraphs. Finally, we provide the unlabeled patients the label of the patient for which the metric is maximum, then add the predicted label to a list. Compare all predictions with the original labels to calculate the accuracy and other metrics.

We experimented with 4 different metrics for classification:

- Nodes: Classifying the label of a patient as the label of the patient with whom this patient has the most number of nodes in the most common subgraphs.
- Edges: Classifying the label of a patient as the label of the patient with whom the patient has the most number of edges in the most common subgraphs.
- Weights: Classifying the label of a patient as the label of the patient with whom the patient has the largest sum of edge weights in the most common subgraphs. An edge weight is equal to the path length from a node to the 'All' node. (See figure 4.3 for an example).
- Nodes with penalty: Classifying the label of a patient as the label of the patient with whom this patient has the most number of nodes in the most common subgraphs divided by the sum of both the source graph and the other graph (graph_2). This correction is necessary, because patients tend to have a larger MCS if one or both graphs that are being compared is large. Thus, giving a bias to large graphs resulting in many false positives.
  The code for the penalty is listed below:

```
(len(mcs.nodes()) / (len(source.nodes())+len(graph_2.
    nodes())))
```

Example of the weights metric in the MCS

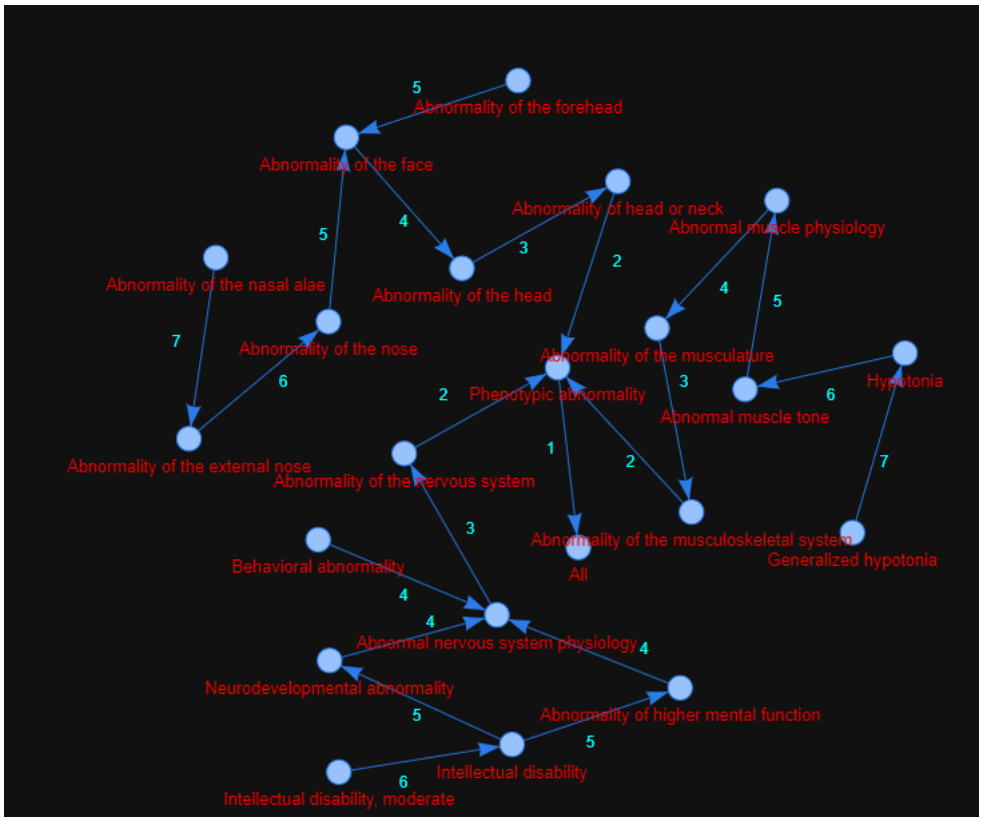


Figure 4.3: The weights metrics is calculated by adding the sum of all edge weights, which is equal to $1+2+2+2+3+3+3+4+4+4+4+4+5+5+5+5+5+6+6+6+7+7=93$ in this MCS.

There are intuitive explanations for the classification metrics:

To begin, the intuition behind the nodes metric is that more nodes in a MCS means that 2 patients share more phenotypic abnormalities. More shared abnormalities make 2 patients more likely to have the same genetic disorder.

In addition, the intuition behind the edge metric is that more edges in a MCS means that 2 patients have more shared phenotypic abnormalities pairs, thus being more likely to have the same genetic disorder. Note that this relates closely to the nodes metric.

Furthermore, the intuition behind the weights metric is that if a MCS has more and longer paths from leaf nodes to the 'All' nodes, thus more shared abnormalities and more shared specific abnormalities, the patients are more likely to have the same genetic disorder. For example, some specific phenotypic abnormalities might belong to a specific genetic disorder or only a few genetic disorders. Therefore, the edges connecting the more specific phenotypic abnormalities have more weight. These nodes have a longer path length to the "All" node. The longer the path length from a node to the "All" node the more specific the phenotypic abnormality is.

To illustrate what a more specific phenotypic abnormality is, we look at the following example from Figure 4.3. For instance, the phenotypic abnormality "generalized hypotonia" is connected to an edge with a weight of 7, which means that it is a more specific phenotypic abnormality than behavioral abnormality, because it is connected to an edge with weight 4.

Finally, the intuition behind the nodes with penalty metric. Larger most common subgraphs implicate that patients have more phenotypic similarities. However, this also offers a bias toward larger HPO graphs. Some graphs have over 200 nodes, while others have 50 nodes. Therefore, some smaller graphs are likely to have a larger MCS between itself and a large graph than itself and a graph that has approximately the same size. Hereto, we provide a penalty to the amount of nodes in the MCS based on the size of the source graph and the second graph.

Next to the classification metrics, a data-analysis technique was used to deal with the nature of the data. To begin, the data is imbalanced. Therefore, we could oversample or undersample the graphs to remove bias from the data. However, oversampling of the HPO graphs will cause overfitting due to the fact that oversampling small classes of 3-20 samples to the majority class, which has 146 samples overfits the data. As a consequence, we used undersampling to reduce the overfitting. We undersampled on 3 classes: *SPOP_1*, *SPOP_2* and *KDM3B*. Moreover, we also undersampled on the 5 largest classes. To be able to undersample on *SPOP_2* we had to remove the classes that are smaller than *SPOP_2* from the data, which is the class *SPOP_1*. Similarly for *KDM3B* we had to remove *SPOP_1* and *SPOP_2*. Moreover, when undersampling the 5 largest classes, the 11 smallest classes had to be removed from the data. This is necessary, because otherwise these classes are not the smallest classes in the data and one always undersamples on the smallest class in the data.

4.0.3 Results for Most Common Subgraph classification

We ran the MCS models with different metrics, namely:

- Number of nodes
- Number of edges
- Sum of weights
- Number of nodes with penalty
- Undersampling *SPOP_1*, *SPOP_2* and *KDM3B*
- Undersampling the 5 largest classes

Some measures are more important than others when analysing the results. The accuracy is not the most important metric because the data is imbalanced which might provide a higher accuracy by guessing the majority class. Therefore, we also look at the confusion matrix and the F1-score.

Furthermore, there are 2 reasons that make the F1-score important. To begin, we want to minimize both false positives and false negatives due to the fact that classifying a patient with a genetic disorder should be as accurate as possible. A false positive occurs when the actual class is predicted as true, while it is false. Conversely, a false negative occurs when the actual class is predicted as false, while it is true. In addition, false positives and false negatives are equally undesired. Precision minimizes the number of false positives and recall minimizes the number of false negatives. The F1-score is the weighted average of precision and recall. Therefore we should put emphasis on the F1-score.

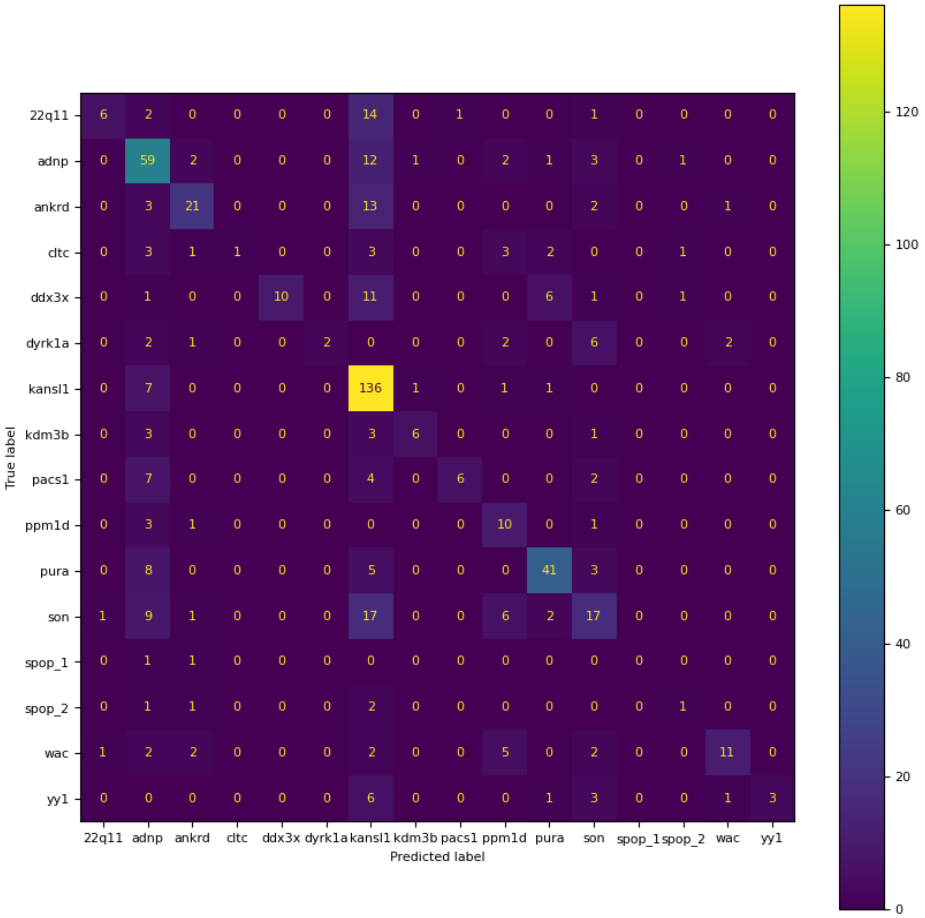
Let us have a look at the results from the models:

Table 4.1: Performance of models

MCS classification model	F1-score	accuracy
Number of nodes	0.47	0.6
Number of edges	0.46	0.6
Sum of weights	0.45	0.57
Number of nodes with penalty	0.68	0.74

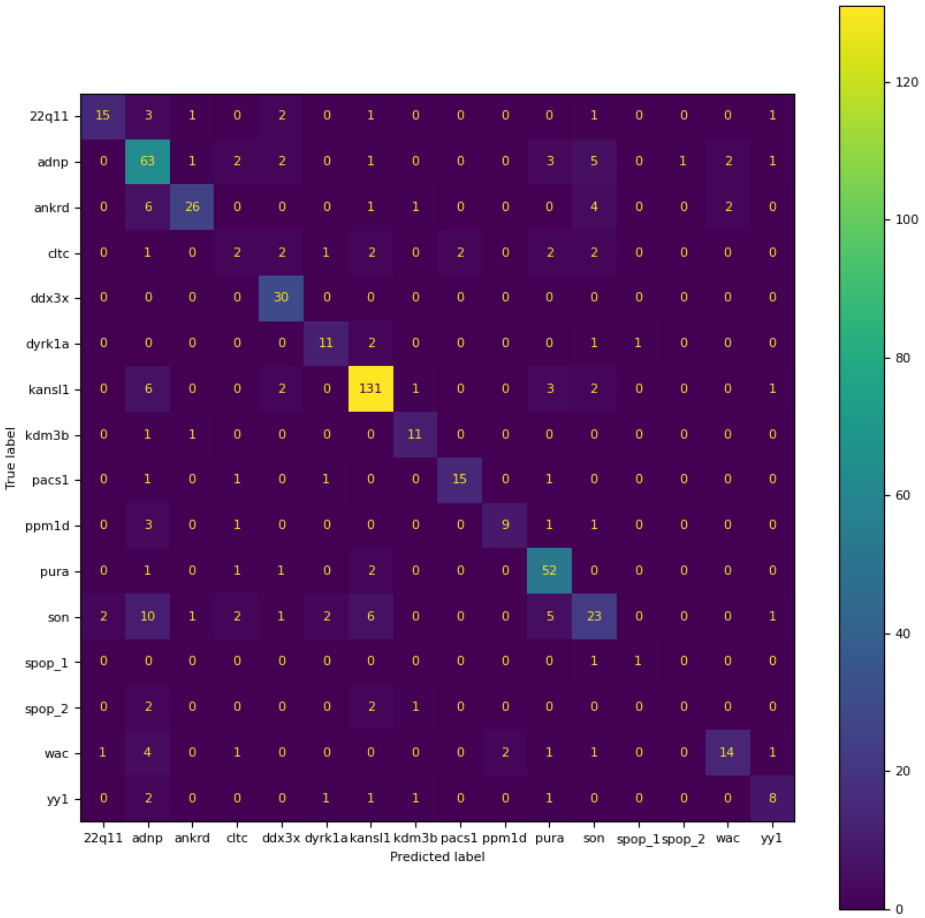
The number of nodes model is the baseline model (a simple model with a decent result). The best performing model is the amount of nodes with penalty which has an accuracy of 74% and a F1-score of 0.68. Let us zoom in on these 2 models and see what differentiates them.

Confusion Matrix for MCS classification based on the number of Nodes



As can be seen from the above image, there are many false positives. Most false positives occur in the 2 largest classes, *KANSL1* and *ADNP*. This is an indication of overfitting. The F1-score is a better measurement for imbalanced data and our F1-score for this model is 0.47. The reason that there are so many false positives in *KANSL1* and *ADNP*, is that *KANSL1* and *ADNP* have some large graphs (200+ nodes), which results in many large most common subgraphs between other classes and *KANSL1*/*ADNP* this in turn presents many false positives.

Confusion Matrix on MCS classification based on number of Nodes with penalty



In contrast to the nodes model, there are significantly less false positives in the majority classes *KANSL1* and *ADNP*. The false positives and false negatives are distributed more evenly among the classes. Moreover, the smaller classes have higher precision and recall. This is due to the fact that we have a correction for the size of the input graphs.

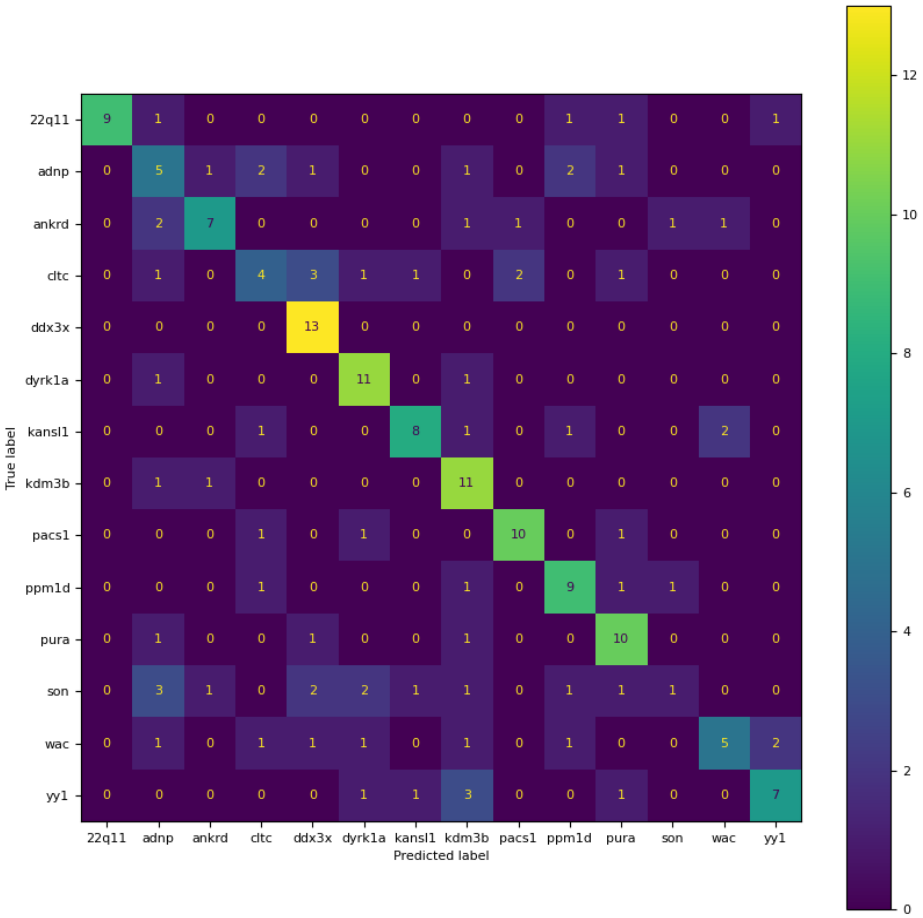
Similarly, let us now have a look at what performance undersampling classes provides. We used the metric that performed best on the whole dataset for undersampling, which is the number of nodes with penalty metric. Note that the F1-score is not that interesting anymore, because it is almost the same as the accuracy due to the fact that the classes are balanced when undersampled.

Table 4.2: Performance of the undersampled models

Undersampling	accuracy	samples per class
SPOP_1	0.34	2
SPOP_2	0.45	5
KDM3B	0.6	13
5 largest classes	0.76	40

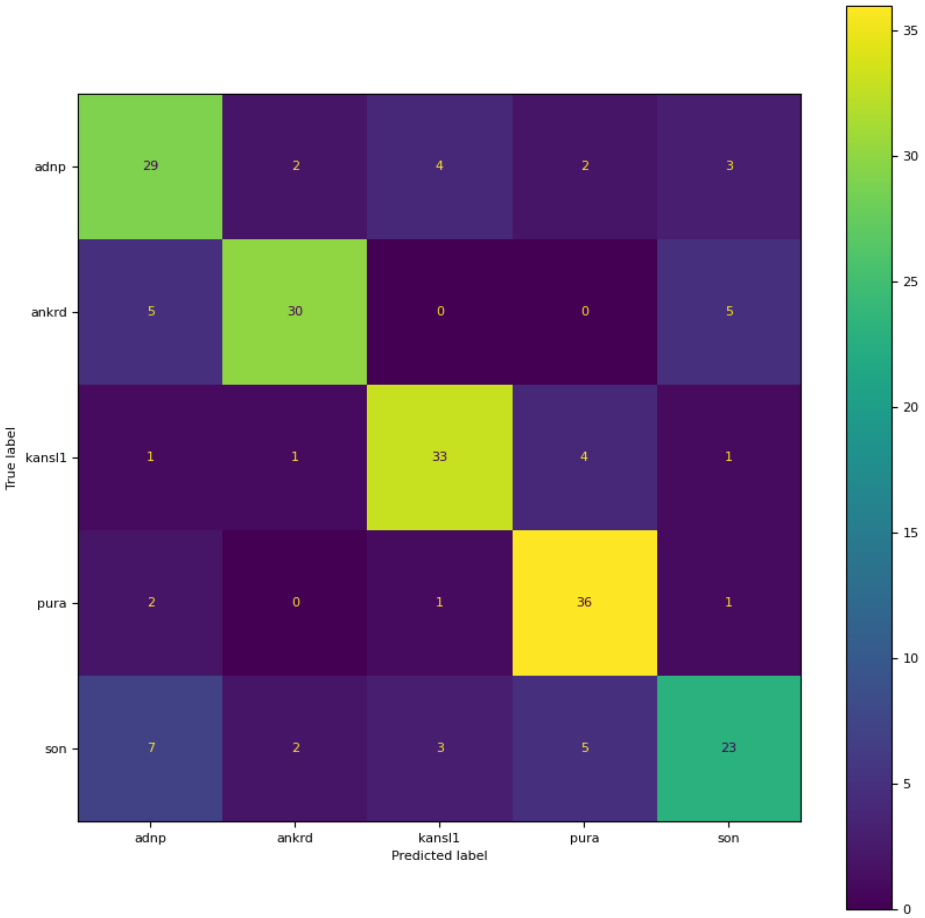
As can be seen from the above table, accuracy is positively correlated with the number of samples per class. Undersampling on *SPOP_1* has 2 samples per class, undersampling on *SPOP_2* has 5 samples per class, undersampling on *KDM3B* has 13 samples per class and when the 5 largest classes are undersampled, there are 40 samples per class. Let us now have a look at the 2 undersampled models with the highest accuracy.

undersampling on *KDM3B*



undersampling on the *KDM3B* class and removing *SPOP_1* and *SPOP_2* from the data has an accuracy of 60%. The false positives and false negatives are distributed evenly among the classes. The within-class accuracy is between 8% (*SON*) and 100% for (*DDX3X*).

Undersampling on the 5 largest classes



Additionally, undersampling on the 5 largest classes has an accuracy of 76%. The gap between the within-class accuracy is smaller than in the previous model, now it is between 58% (*SON*) and 90% (*PURA*). This is due to the fact that the samples per class are higher, 40 instead of 13.

4.0.4 Kernel-based graph classification

Kernel-based graph classification is grounded on the computation of a kernel value between all graphs. We used the package GraKeL [10] for kernel-based graph classification, which provides implementations for numerous graph kernels. We used the graph kernels in combination with support vector machines for classification. This is the global design for kernel-based graph classification.

We built a model for kernel-based graph classification which includes the following steps:

- Initialize a kernel from GraKeL
- Initialize either KFold or stratified KFold
- Convert the networkx graphs to GraKeL graphs
- Compute the kernel matrix
- Initialize a support vector machine with the precomputed kernel
- Split the set into train and test
- Train (fit) and test (predict) the model
- Calculate the accuracy and other metrics

KFold versus stratified KFold

KFold and stratified KFold have advantages and disadvantages. It is important to choose between KFold or stratified KFold. The advantage of stratified KFold is that it represents the data as balanced as possible in the train set. The larger the K, the longer the training and testing takes, however there is less bias when using a large K. leave-one-out cross-validation provides the best accuracy and F1-score, but is computationally most expensive, because K is equal to the number of input graphs. Stratified KFold presents worse accuracy but is faster.

To illustrate what this can mean in practice, we ask the following question: What is the difference in accuracy when using stratified KFold instead of leave-one-out cross validation? It depends on the K that one chooses for stratified KFold. To begin, when k=2 (the number of patients in the minority class), the accuracy is 63%. Furthermore, for k=5, the accuracy is 75%. In addition, for k=10, the accuracy is 77%. Lastly, for k=16 (the number of classes), the accuracy is 79.5%. The accuracy is 80.3% when using leave-one-out cross validation. Concluding, the difference in accuracy, depends on the value of k. It varies between -17% for k=2 and -0.8% for k=16, the latter difference in accuracy is negligible given our data.

All things considered, we ask the paramount question: When should one run stratified KFold instead of leave-one-out? To begin, given our data, leave-one-out cross-validation and stratified KFold are both fast, because we only have 553 graphs. In contrast, if one would have 100.000 graphs in the data, leave-one-out cross-validation might become infeasible to run. As mentioned before, leave-one-out cross-validation gives higher accuracy than stratified KFold. Therefore, one should run leave-one-out cross-validation as long as it is practical to run.

We used the following kernels from GraKeL for classification:

- Weisfeiler Lehman Optimal Assignment
- Vertex Histogram
- Edge Histogram
- shortest path

We chose the kernels based on the practitioner's guide in (figure 10) of [6]. To begin, the vertex attributes are important, the graphs are not necessarily large, and the global structure is important. Resulting in a recommendation on prioritizing optimal assignment Weisfeiler-Lehman kernels, edge label kernels, vertex label kernels and, shortest path kernels.

In addition, some recommended kernels were excluded from the results. The guide also recommends using sampling methods, lovász and, long random walk kernels. We chose to exclude these in the results, due to computational constraints.

Next to the kernels that had to be excluded, an addition to the graphs was necessary in order to run a kernel that was included. The edge histogram kernel requires edge labels in the graphs. Therefore we added the path length to the edge of a node, based on the path length from the node to the "All" node. The edge weights are chosen in this manner to indicate that deeper edges connect more specific phenotypic abnormalities and therefore have more weight (same weights as the weights metric in MCS classification).

Now that we know which graph kernels to use, we can look at the most interesting graph kernels. Surprisingly 3 out of the 4 graph kernels reveal similar results, therefore we have 3 reasons to explain 2 graph kernels explicitly. First of all, we chose to explain the vertex histogram kernel, because it is a simple baseline kernel. Secondly, we chose to explain the Weisfeiler-Lehman optimal assignment kernel, because it is a state-of-the-art and sophisticated kernel that is known for improving classification results over baseline kernels [5]. Finally, the simple vertex histogram kernel and the sophisticated Weisfeiler-Lehman optimal assignment kernel have the same performance given the data, which instigates further investigation.

To begin, the vertex histogram kernel functions as follows. The VH-kernel calculates a kernel value between all pairs of a graph. Each graph receives a feature vector which consists of the amount of times a node label (HPO term) occurs in that graph, which is always 0 or 1 in HPO graphs. This is due to the fact that a specific phenotypic abnormality occurs at most once in a graph. After calculating the feature vectors for the 2 graphs that are being compared, it takes the dot product between the feature vectors, which results in the kernel value. This process is illustrated by an example in Figure 4.4.

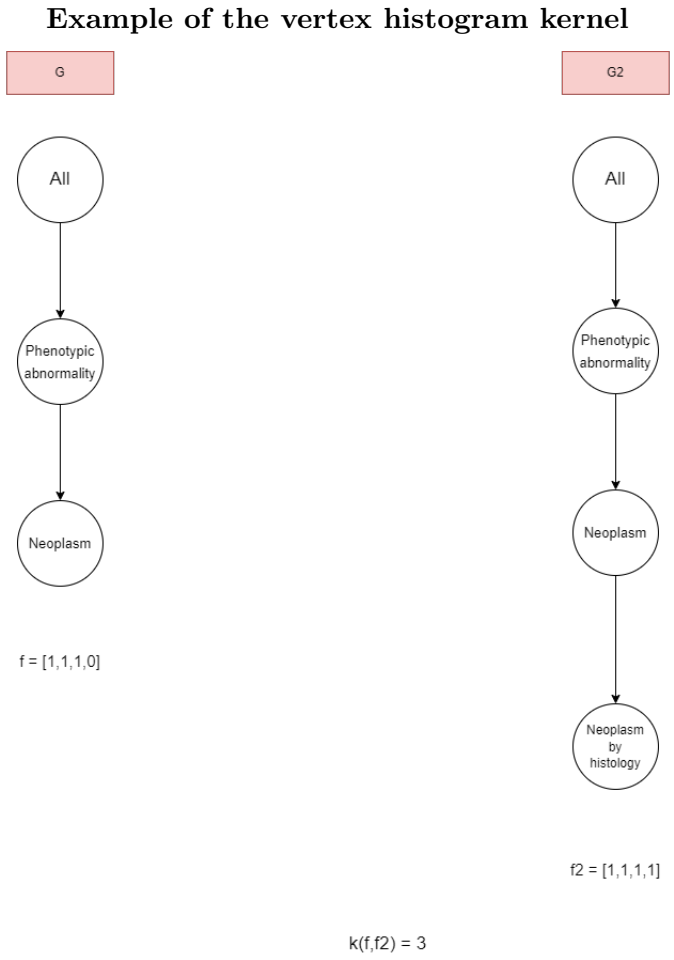
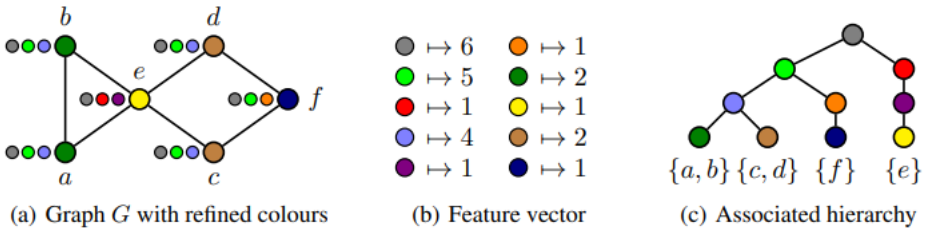


Figure 4.4: The kernel value between graph G and graph G2 is calculated as followed: each graph has a feature vector, respectively f and f2, which indicates how often a HPO term occurs in that graph. In this example, the feature vector can be interpreted like this: f = [amount of occurrences of HPO term 'All' in the graph, amount of occurrences of HPO term 'phenotypic abnormality' in the graph, amount of occurrences of HPO term 'neoplasm' in the graph, amount of occurrences of HPO term 'neoplasm by histology' in the graph]
Now we complete the feature vectors for graph G and G2. For graph G, it is f = [1,1,1,0] and for graph G2, f2 = [1,1,1,1]. Lastly, we can calculate the kernel value by taking the dot product of the 2 feature vectors which is equal to k(f,f2) = $1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 = 3$

Furthermore, the Weisfeiler-Lehman optimal assignment kernel functions as follows. The kernel is based on iterative vertex color refinement. For a parameter h and a graph G with initial labels τ, a sequence $(\tau_0, \ldots, \tau_h)$ of refined labels, referred to as colors is computed, where $\tau_0 = \tau$ and τ_i is obtained from τ_{i-1} by the following procedure:



(a) Graph G with refined colours (b) Feature vector (c) Associated hierarchy

Figure 4.5:

Sort the multiset of colors $\tau_{i-1}(u) : vu \in E(G)$ for every vertex v lexicographically to obtain a unique sequence of colors and add $\tau_{i-1}(v)$ as first element. Assign a new color $\tau_i(v)$ to every vertex v by employing a one-to-one mapping from sequences to new colors, 4.5a of the figure illustrates the refinement process. 4.5b reveals the color refinement process over time, first all 6 vertices are gray, after the first iteration 5 vertices are green and 1 is red, and so on. The kernel value for the Weisfeiler-Lehman optimal assignment can be calculated by taking the histogram intersection of the feature vectors of the graphs [5].

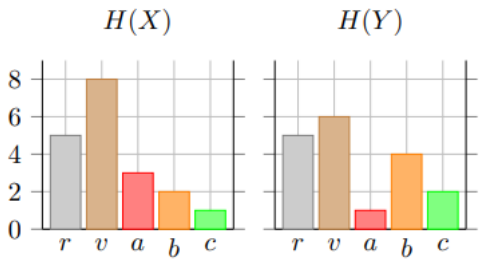


Figure 4.6: The histogram intersection between the 2 feature vectors of H(X) and H(Y) is calculated as followed: take the minimum value per color and sum these numbers to receive the kernel value. For example the minimum value for the gray columns with letter r is 5. min{5,5}+min{8,6}+min{3,1}+min{2,4}+min{1,2}=15

4.0.5 Results of kernel-based graph classification

Before diving into the results, there are 3 considerations that have to be made when choosing a model. To begin, we select a kernel. Additionally, we take a value for K. Then, we choose KFold or stratified Kfold. We always use normalization.

In addition, how does normalization operate? Normalization divides the kernel value between 2 graphs by the square root of the multiplication of the kernel value between graph 1 and graph 2 and the kernel value between graph 2 and itself. This results in the following formula: $\frac{\mathbf{k}(\mathbf{G1}, \mathbf{G2})}{\sqrt{(\mathbf{k}(\mathbf{G1}, \mathbf{G2}) * \mathbf{k}(\mathbf{G2}, \mathbf{G2})}}$.

Thus, the kernel value between a graph and any other graph will be between 0 and 1 and between itself it will be 1.

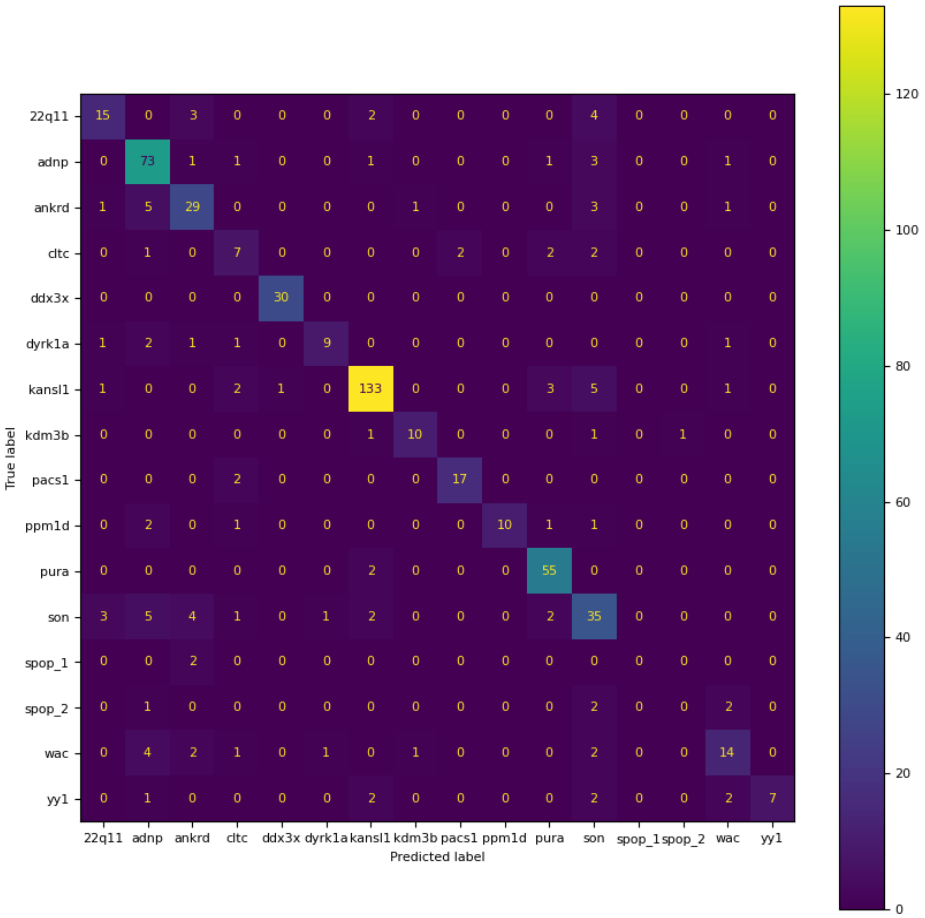
Let us have a look at the overview of the accuracy of all kernels used:

Table 4.3: Performance of the graph kernels

Graph Kernel	F1-score	accuracy
Weisfeiler Lehman Optimal Assignment	0.75	0.8
Vertex Histogram	0.75	0.8
Edge Histogram	0.24	0.31
Shortest Path	0.75	0.81

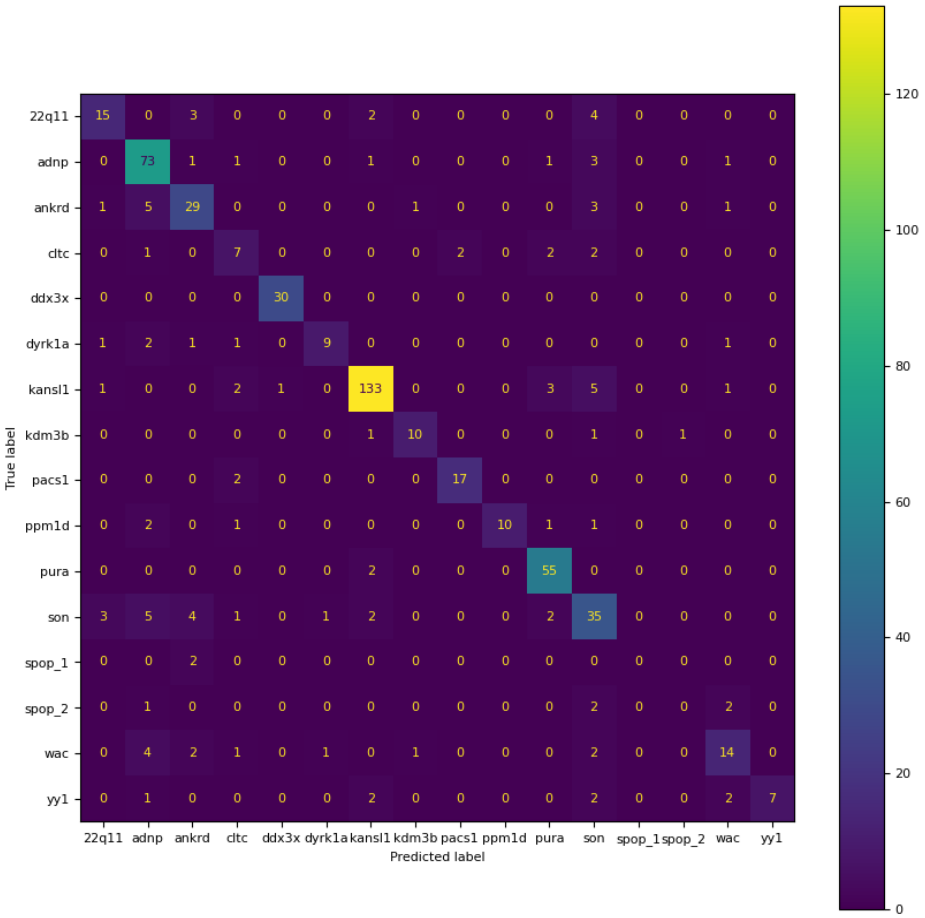
As mentioned before, let us have a look at 2 interesting graph kernels that reveal similar results. The state-of-the-art Weisfeiler-Lehman optimal assignment kernel and the simple vertex histogram kernel.

Weisfeiler lehman optimal assignment kernel classification



Weisfeiler-Lehman optimal assignment kernel (state-of-the-art) with normalization, the whole data set and leave-one-out cross validation has an accuracy of 80% and a F1-score of 0.75.

vertex histogram kernel classification



The vertex histogram kernel, on the whole dataset and leave-one-out cross validation has an accuracy of 80% and the F1-score is 0.75. We receive the same results for this simple graph kernel and the sophisticated Weisfeiler-Lehman optimal assignment kernel. In the section "inspection of models", we will discuss the causes of these similar results.

4.0.6 Inspection of models

Table 4.4: Overview of all models

Model	F1-score	accuracy
MCS classification: Number of nodes	0.47	0.6
MCS classification: Number of edges	0.46	0.6
MCS classification: Sum of weights	0.45	0.57
MCS classification: Number of nodes with penalty	0.68	0.74
Graph Kernel: Weisfeiler Lehman Optimal Assignment	0.75	0.8
Graph Kernel: Vertex Histogram	0.75	0.8
Graph Kernel: Edge Histogram	0.24	0.31
Graph Kernel: Shortest Path	0.75	0.81

What differentiates the MCS models? The baseline MCS classification number of nodes model performs approximately as well as the MCS classification number of edges and MCS classification sum of weights models. Thus, classifying based on the sum of weights/ number of edges does not improve classification. The MCS classification number of nodes with penalty does significantly improve the baseline model, it improves the nodes baseline model by 14% accuracy and 0.17 F1-score. This is due to the fact that this model corrects the bias such that larger graphs should not necessarily have larger most common subgraphs.

Likewise, there is an explanation for the difference in performance between the edge histogram kernel and other kernels. The simple edge histogram kernel that uses edge weights has an accuracy of 31%, because it mostly predicts *KANSL1* (the majority class) due to the fact that *KANSL1* has the largest graphs and therefore also receives large kernel values between other graphs and *KANSL1* graphs. This results in many false positives.

Sophisticated versus Simple graph kernels

Why are the accuracy and F1-score of the simple vertex histogram kernel and the sophisticated Weisfeiler-Lehman optimal assignment kernel (WLOA) the same? This is because the HPO graphs do not have much information and

the order of the nodes is typically the same due to the hierarchical structure of the HPO. A node typically has the same parent, but a parent node can have different child nodes. The sophisticated kernels can not exploit the graphs, because the node labels and directed acyclic graph structure is everything that exists in the graphs. There are no extensive node and/or edge attributes. The vertex histogram kernel value between 2 graphs is the sum of HPO terms that the graphs share. While the WLOA involves sophisticated techniques, such as the iterative color refinement process, which is harder to understand and makes an intuitive explanation of the results hard. The results of both kernels are the same, thus we can conclude that the WLOA can not extract more useful information than the sum of shared nodes between graphs as the kernel value.

Next to the differences between models it is important to investigate models by itself, therefore we ask the following question: Which properties of the graphs are most important for each classification method in predicting the genetic disorder of patients?

MCS classification

There are 2 import properties for predicting the genetic disorder of a patient for each MCS method. To begin, the first property is the pairs of shared phenotypic abnormalities between graphs. In addition, the second property is the metric that is used to decide which MCS is most similar to the graph of the unlabeled patient. The number of nodes in the MCS is the most important property in predicting the genetic disorder for the nodes metric. The number of edges in the MCS for the edges metric. The sum of edge weights in the MCS for the weights metric. The penalty metric depends on the number of nodes in the original graphs and MCS.

Kernel-based classification

We ran a total of 4 different kernels from the GraKeL package. Some kernels focus on certain properties about the nodes and other kernels focus on edge labels between graphs. Let us have a look at 2 distinct graph kernels. These kernels are the vertex and edge histogram. The vertex histogram kernel calculates the sum of shared nodes between 2 graphs, so the most important properties are the node labels. The edge histogram kernel calculates the number of occurrences of edge weights in graphs. For each graph, it creates a feature vector in which the amount of times a weight occurs is denoted. Then the kernel value is calculated by taking the inner product of the feature vectors of the 2 graphs. Thus, the most important properties are the edge labels.

In conclusion, we ask the following question: Which model performed best in classifying the genetic disorder of patients? We should look at a combination of the F1-score and the simplicity of the model to find the best performing model. The vertex histogram kernel and Weisfeiler-Lehman optimal assignment have the same F1-score of 0.75. However, the vertex histogram kernel is a kernel that is easier to understand, explain, and less complicated than the Weisfeiler-Lehman optimal assignment kernel. Therefore, the best performing model for classification is the vertex histogram kernel.

Next to being the best performing model, it could be used in a real-life setting. The model has a high F1-score given the data, therefore it could be used in practice as a tool for clinical geneticists to predict what gene deficit a patient with a VUS has. However, this tool can only be used if the suspected gene deficit is in the data.

Chapter 5

Conclusions

The purpose of this research was to determine if it is possible to accurately classify which genetic disorder a patient has based on HPO graph similarity classification. Based on the MCS classification models and graph kernel classification models, we can conclude that it is possible to accurately classify which genetic disorder a patient has given the data.

Can the hypothetical case of the patient Paul be solved? The clinical geneticist named Leonardo knows that the VUS is in the *KANSL1* gene. He creates a HPO graph for Paul his phenotypic abnormalities. Leonardo expects that if the *KANSL1* mutation causes the phenotypic abnormalities, then when you do binary classification on a model (e.g. create a graph kernel model for binary classification that is trained on *KANSL1* and control patients) that is trained on 146 *KANSL1* patients (The amount of *KANSL1* patients in our data) and 146 control patients (patients with a different genetic disorder) the predicted class would be *KANSL1*. Leonardo runs the model with Paul's HPO graph and the predicted result is KANSL1 (1 in binary classification). Hereto, the clinical geneticist can explain the effects of the mutation to Paul and his family, improve treatment for Paul, and give closure to Paul and his family.

Next to the main conclusion, at least four other conclusions can be drawn from the models. To begin, the best performing model given the data is the vertex histogram kernel when using leave-one-out cross validation and normalization. Furthermore, the most important property in accurately predicting the genetic disorder of a patient are the node labels in the graphs. Moreover, the simple graph kernel (vertex histogram) performs approximately as well as the sophisticated graph kernels (WLOA, shortest path) given the data. Lastly, this model could be used as a tool for clinical geneticists to predict what gene deficit a patient with a VUS has. However, this tool can only be used if the suspected gene deficit is in one of the genes

included in the data.

On top of these findings, undersampling also provides some insights. Undersampling the graphs provides an explanation that even if the data is imbalanced, the models are not biased towards the majority class, this is due to the fact that the accuracy when undersampling on the 5 largest classes is higher than the accuracy when there is no undersampling. We would expect a lower accuracy for undersampling on the 5 largest classes if the model is biased towards the majority class. Moreover, the false positives/negatives are evenly distributed when undersampling on the 5 largest classes, which is an indication that there is no bias towards any of these classes. Additionally, when you undersample, the more HPO graphs per class the higher the accuracy, this can be seen in Table 4.2.

The findings of this study have to be seen in light of some limitations, there are two limitations that could be addressed in future research. To begin, the data is limited as only 16 out of thousands of genetic disorders are in the data. In addition, we only have data of 553 patients and the graphs only include HPO term data and the hierarchical structure between the HPO terms.

Next to these limitations, we have 3 recommendations for future research. First of all, there are some interesting features that could be added to the HPO graphs: adding the amount of gene associations and disease associations to the node labels, these numbers can be found in the HPO. Secondly, adding a value to the nodes for how rare the phenotypic abnormality is. Finally, adding this numerical information to the node labels in the graphs makes it possible to build a convolutional graph neural network for classification. This is impossible otherwise, because graph neural networks require feature vectors for the nodes in the graph.

The results corroborate the findings of existing literature in other domains. Akin to [1], [2], [3], [4], [11] most common subgraph similarity classification and kernel-based graph classification has shown promising results in the domain of clinical genetics. The practitioner's guide in [6] proved useful in choosing the right graph kernels for the given problem.

On the other hand, there are some interesting differences between our HPO-based classification and HPO-based classification in existing literature. Contrary to Köhler *et al.* [7] we classified patients with one disease instead of giving a list of candidate diseases ranked by similarity. Moreover, we compared HPO graphs of patients to discover similarities between patients, while they compared HPO terms of patients with disease vectors (all HPO terms characterizing a disease) to discover similarities between a patient and the diseases in the database. In short, we classified based on the similarity be-

tween patients and they classified based on the similarity between a patient and diseases.

In addition, a comparison between The Phenomizer of Köhler *et al.* [7] and our models was made. Sasja and I wrote a Python script in which we used the query_phenomizer Python package to run the Phenomizer with our patient data. Note that the differences mentioned above still exist. Moreover, the data consists of 553 patients that have one of 16 genetic disorders and a patient is therefore only compared to 552 other patients that can have 1 out of 16 genetic disorders, while The Phenomizer has thousands of disease vectors to which a patient is compared. If the patient's disease is in the top 100 of candidate diseases from the output of the Phenomizer it was counted as correctly classified to correct somewhat for the significant difference mentioned in the sentence above. Sasja and I also added the Online Mendelian Inheritance in Man (OMIM) names for the genetic disorders that are in the data to make sure that if the genetic name is not in the candidate diseases, but the OMIM name is, it is still classified as correct. The Phenomizer ran on the data and classified a patient as correctly classified if the correct disease was in the top 100 candidate diseases, this resulted in an accuracy of 4.2%. This is significantly lower than the 80% accuracy that was obtained with the Vertex Histogram model. There are several explanations for these differences. First of all, the accuracy is lower due to the fact that The Phenomizer compares the patient to thousands of disease vectors, while the MCS and graph kernel models compare it to 552 patients that have one of 16 genetic disorders. Second of all, patient-patient comparison, which happens with MCS and graph kernel classification, seems to provide superior results, while patient to disease vector comparisons for classification seem to perform worse given the results of The Phenomizer.

Patient-to-patient similarity classification based on the HPO graphs reveals promising results given our data. However, this is not necessarily the case for all genetic disorders. Like mentioned by Ratnaike *et al.* [8] mitochondrial disorders are hard to distinguish from non-mitochondrial disorders due to much overlap in clinical phenotypes. Hereto, the results might depend on the genetic disorders that are included in the data.

Concluding, we can accurately classify which genetic disorder a patient has based on HPO graph similarity classification.

Chapter 6

Data Availability

The data are not publicly available due to the General Data Protection Regulation (EU GDPR) restrictions. However, the code is freely available at: `https://github.com/DatSplit/HPO-Graph-Similarity-classification`.

Bibliography

[1] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, *Protein function prediction via graph kernels*, Bioinformatics, 21 (2005), pp. i47–i56.

[2] Y. Cao, T. Jiang, and T. Girke, *A maximum common substructure-based algorithm for searching and predicting drug-like compounds*, Bioinformatics, 24 (2008), pp. i366–i374.

[3] X. Cui, J. Xiang, H. Guo, G. Yin, H. Zhang, F. Lan, and J. Chen, *Classification of alzheimer's disease, mild cognitive impairment, and normal controls with subnetwork selection and graph kernel principal component analysis based on minimum spanning tree brain functional network*, Frontiers in Computational Neuroscience, 12 (2018), p. 31.

[4] B. Jie, M. Liu, X. Jiang, and D. Zhang, *Sub-network based kernels for brain network classification*, in Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '16, New York, NY, USA, 2016, Association for Computing Machinery, p. 622–629.

[5] N. M. Kriege, P. Giscard, and R. C. Wilson, *On valid optimal assignment kernels and applications to graph classification*, CoRR, abs/1606.01141 (2016).

[6] N. M. Kriege, F. D. Johansson, and C. Morris, *A survey on graph kernels*, Applied Network Science, 5 (2020).

[7] S. Köhler, M. H. Schulz, P. Krawitz, S. Bauer, S. Dölken, C. E. Ott, C. Mundlos, D. Horn, S. Mundlos, and P. N. Robinson, *Clinical diagnostics in human genetics with semantic similarity searches in ontologies*, The American Journal of Human Genetics, 85 (2009), pp. 457–464.

[8] T. E. Ratnaike, D. Greene, W. Wei, A. Sanchis-Juan, K. R. Schon, J. van den Ameele, L. Raymond, R. Horvath, E. Turro,

and P. F. Chinnery, *MitoPhen database: a human phenotype ontology-based approach to identify mitochondrial DNA diseases*, Nucleic Acids Research, 49 (2021), pp. 9686–9695.

[9] P. N. Robinson, S. Köhler, S. Bauer, D. Seelow, D. Horn, and S. Mundlos, *The human phenotype ontology: A tool for annotating and analyzing human hereditary disease*, The American Journal of Human Genetics, 83 (2008), pp. 610–615.

[10] G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis, *Grakel: A graph kernel library in python*, Journal of Machine Learning Research, 21 (2020), pp. 1–5.

[11] S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi, *Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity*, Bioinformatics, 21 (2005), pp. i359–i368.