Bioinformatics, 38, 2022, ii120–ii126
https://doi.org/10.1093/bioinformatics/btac478
Proceedings of ECCB2022

GNN-SubNet: disease subnetwork detection with explainable graph neural networks

Bastian Pfeifer[1,*], Anna Saranti[1] and Andreas Holzinger[1,2,3]

[1]Institute for Medical Informatics Statistics and Documentation, Medical University Graz, Graz, Austria, [2]Human-Centered AI Lab, Department of Forest- and Soil Sciences, University of Natural Resources and Life Sciences Vienna, Vienna, Austria and [3]Alberta Machine Intelligence Institute, University of Alberta, Edmonton, Canada

*To whom correspondence should be addressed.

Abstract

Motivation: The tremendous success of graphical neural networks (GNNs) already had a major impact on systems biology research. For example, GNNs are currently being used for drug target recognition in protein–drug interaction networks, as well as for cancer gene discovery and more. Important aspects whose practical relevance is often underestimated are comprehensibility, interpretability and explainability.

Results: In this work, we present a novel graph-based deep learning framework for disease subnetwork detection via explainable GNNs. Each patient is represented by the topology of a protein–protein interaction (PPI) network, and the nodes are enriched with multi-omics features from gene expression and DNA methylation. In addition, we propose a modification of the GNNexplainer that provides model-wide explanations for improved disease subnetwork detection.

Availability and implementation: The proposed methods and tools are implemented in the GNN-SubNet Python package, which we have made available on our GitHub for the international research community (https://github.com/pievos101/GNN-SubNet).

Contact: bastian.pfeifer@medunigraz.at

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Graph neural networks (GNNs) have attracted much attention in general (Scarselli *et al.*, 2009; Wu *et al.*, 2020), in bioinformatics (Zhang *et al.*, 2021) and biomedical research in particular (Zhou *et al.*, 2020).

Recently, significant research efforts have been made to apply deep learning (DL) methods to graphs (Bacciu *et al.*, 2020). This progress resulted in tremendous advances in graph analysis techniques, which are useful in many biomedical applications (Zhang *et al.*, 2020), e.g. for protein–drug interaction detection (Zitnik *et al.*, 2018). A very recent work presents a graph-based framework for detecting novel cancer genes by using GNNs for node classification (Schulte-Sasse *et al.*, 2021). The authors label genes of a protein–protein interaction (PPI) network according to their cancer relevance (relevant or not). DL-based node classification is applied to predict whether or not an unlabeled protein is relevant to cancer.

A key feature of GNNs is that they seemingly allow for the integration of knowledge graphs, (Ji *et al.*, 2021) into the algorithmic pipeline, such as ontologies (Kulmanov *et al.*, 2020) and/or PPI networks (Jeanquartier *et al.*, 2015; Liu *et al.*, 2009). This feature enables a domain expert to integrate human experience, human conceptual knowledge and contextual understanding into the machine learning architectures. Such a human-in-the-loop (expert-in-the-loop) can sometimes—of course not always—help to obtain more robust, reliable and also more interpretable results (Holzinger *et al.*, 2019). It should be emphasized that robust, explainable, and trustworthy solutions are among the major goals of the artificial intelligence (AI) community for the near future (Holzinger *et al.*, 2021a).

Such solutions are of practical relevance in critical areas where we suffer from low data quality, especially where we just do not have the i.i.d. data we actually need. Therefore, the use of AI in areas that impact human life (e.g. agriculture, climate, health…) has led to an increased demand for trustworthy AI. This is especially true in sensitive areas such as biomedicine, where traceability, transparency and interpretability are not ends in themselves, but are now even mandatory due to regulatory requirements Finally, the 'why' Pearl (2019) is often more important to science than a pure result. Consequently, both explainability and robustness can promote reliability and trust and ensure that humans remain in control and thus that human intelligence is supported by AI and by no means replaced.

In our work, we place a particular emphasis on the integration of PPI networks for disease subnetwork detection. Most existing methods for disease subnetwork detection rely on unsupervised clustering and/or community detection algorithms (Choobdar *et al.*, 2019) to detect modules with correlated node features. We believe that functional subnetworks with high classification accuracy, where node features are not necessarily correlated, may contain an additional set of biologically relevant disease modules. While conventional feature selection techniques can be used for this task, most of them are not directly applicable to graph-structured data. An exception is our proposed method (Pfeifer *et al.*, 2021), where we introduce a greedy decision forest for subnetwork detection. To demonstrate the applicability of this approach, we enriched the nodes of a PPI network with multi-omic features. Decision trees are derived from this network using random walks. The decision trees evolve on this network to a minimal set of high-performance subnetworks. In this work, while we pursue a similar research goal, we further use powerful graph DL architectures and explainable AI methods (Holzinger *et al.*, 2021b) for DL-based disease subnetwork detection. To the best of our knowledge, this is novel and thus represents the first work that uses explainable AI for disease subnetwork discovery.

This article is organized as follows: first, a brief summary of the proposed methodology for disease subnetwork detection is given. Section 3 describes the methodology, the GNN methods used and the validation data in detail. Section 4 presents the results obtained using synthetic datasets as well as multimodal human cancer data. Section 5 discusses the work presented and possible future research directions.

2 New approach

In this work, we propose explainable GNNs for the detection of disease subnetworks. We have formulated the subnetwork detection task as a graph classification problem, where graph topologies are the same for all instances, but with varying node feature values. The following methodology is presented.

Each patient is represented by the graph topology of a PPI network, where proteins are reflected by the nodes and the edges indicate a functional relationship between these proteins. The nodes of the patient-specific graphs are enriched by multi-omic feature values, such as mRNA gene expression and DNA methylation. Following, we perform graph classification in order to classify patients into a cancer-specific group and a randomized cancer group (Fig. 1). As a consequence, a GNN model is trained on domain-knowledge-induced trajectories specified by the PPI network, which overall may result in more reliable and interpretable outcomes (Tiddi and Schlobach, 2022).

In order to ultimately uncover the decisions of the GNN classifier, we utilize a modified version of the GNNexplainer algorithm, by optimizing a *model-wide* node feature mask (see Section 3 for details). From the obtained node importance values, we compute edge relevance scores. We assign these values as edge-weights to the PPI Network and apply weighted community detection algorithms. The detected communities with high edge importance scores represent the potential disease subnetworks.

3 Materials and methods

3.1 GNN architecture

We have employed a GNN classifier as evaluated and implemented by Xu *et al.* (2018). The authors propose a graph isomorphism network (GIN) architecture that has been proven to have better classification performance than other GNN architectures. One of the first GNN architectures that was invented was dealing with the graph in a similar way as a CNN processes images or any kind of typical grid-structured data (Kipf and Welling, 2016). In the same way that CNN filters are convoluted with a portion of the input, the same applied to the graph convolutional networks (GCN). The main difference lies in the fact that in GCNs the portion of the input is a subgraph (k-hop neighborhood) *around* a central node, whereas in a CNN the neighborhood of the central element is structured as a grid.

In Figure 2, the CNN's and GNN's basic aggregation operation involving information from the neighborhood is depicted. Mathematically, this can be described by the following operation:

$$a_v^{(k)} = \mathrm{AGGREGATE}^{(k)}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}) \tag{1}$$

where k is the number of aggregation iterations and equals to the number of hops of the neighborhood $\mathcal{N}$ that will be considered. The aggregation operation uses all features of the neighboring nodes (denoted with u); those can be of any form and can numerically encode several characteristics like size, color, shape and so on. As CNNs that process images typically aggregate three values (RGB) of each neighboring pixel, the GNN will aggregate any number of features representing any feature selected by the domain expert and the data scientist. Those features are denoted with h and are also called embeddings.

After the aggregation operation, the combine operation provides the new values for the features of node v:

$$h_v^{(k)} = \mathrm{COMBINE}^{(k)}(h_v^{(k-1)}, a_v^{(k)}) \tag{2}$$

Those two operations, namely aggregate and combine are performed several times. The initial values of the features are replaced with new, informed ones that help the task at hand. Typically, GNNs can be used for node classification, link prediction and graph classification. Node as well as graph classification use the end values of the node features after the last application of aggregate and combine. Until now, the way aggregate and combine are implemented is not fully addressed. The underlying operation in the GCN (Kipf and Welling, 2016) is an element-wise mean pooling followed by a rectified linear unit non-linear activation function. The researchers that invented GIN (Xu *et al.*, 2018) have proven that aggregations that are implemented by the `mean()` and `max()` function cannot distinguish between very simple graph structures; therefore, they are not adequate for computing embeddings, especially when the task is graph classification.

As far as the combining step is concerned, the GIN architecture learns a function with the use of multilayer perceptrons (MLPs). This provides the necessary flexibility for injectiveness, maximum possible discrimination ability as well as the property of mapping similar graph structures to nearby embeddings. The overall equation of aggregation and combine steps in GIN is provided by the following Equation (3):

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left((1 + e^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}\right) \tag{3}$$

More information about the derivation of Equation (3), as well as theoretical basis can be found in Xu *et al.* (2018).

The exact architecture that was used for the proposed application consists of three MLPs. Each of those perceptrons is preceded with a pooling layer and succeeded with a batch normalization layer. Following those, there are five fully connected layers. Within the fully connected layers, each neuron is connected to all neurons in the layer before and after that layer. The MLPs consist of three layers, of which two are fully connected layers and layer for batch normalization layer.

3.2 Explainable AI for disease subnetwork detection

In recent years, parallel to the development of different GNN architectures, several strategies were invented to explain their decision process. Most of them are built on the assumption that a part of the input was the most important for the prediction in a similar way that the explanation for an image classification CNN will point out the areas in the image that were decisive and will ignore the ones that contain background.

Explainable AI methods usually search for relevant subgraphs and their motifs (Luo *et al.*, 2020; Ying *et al.*, 2019), walks (Schnake *et al.*, 2020) or even try to create probabilistic graphical models (Koller and Friedman, 2009; Saranti *et al.*, 2019) which are causal structures, out of counterfactual examples that are computed by informed optimization problems (Vu and Thai, 2020).

GNNExplainer is used to compute the important subgraph G_S of the computation graph G_c of an input graph G that is going to be explained. This is achieved by graph masking as well as node feature masking, where the goal is to learn to mask the relevant part of the computation graph as well as the decisive node features. Those masks are found by an optimization algorithm that iteratively tries to find the substructure that maximizes the mutual information w.r.t. the prediction score. Equation (4) shows the optimization

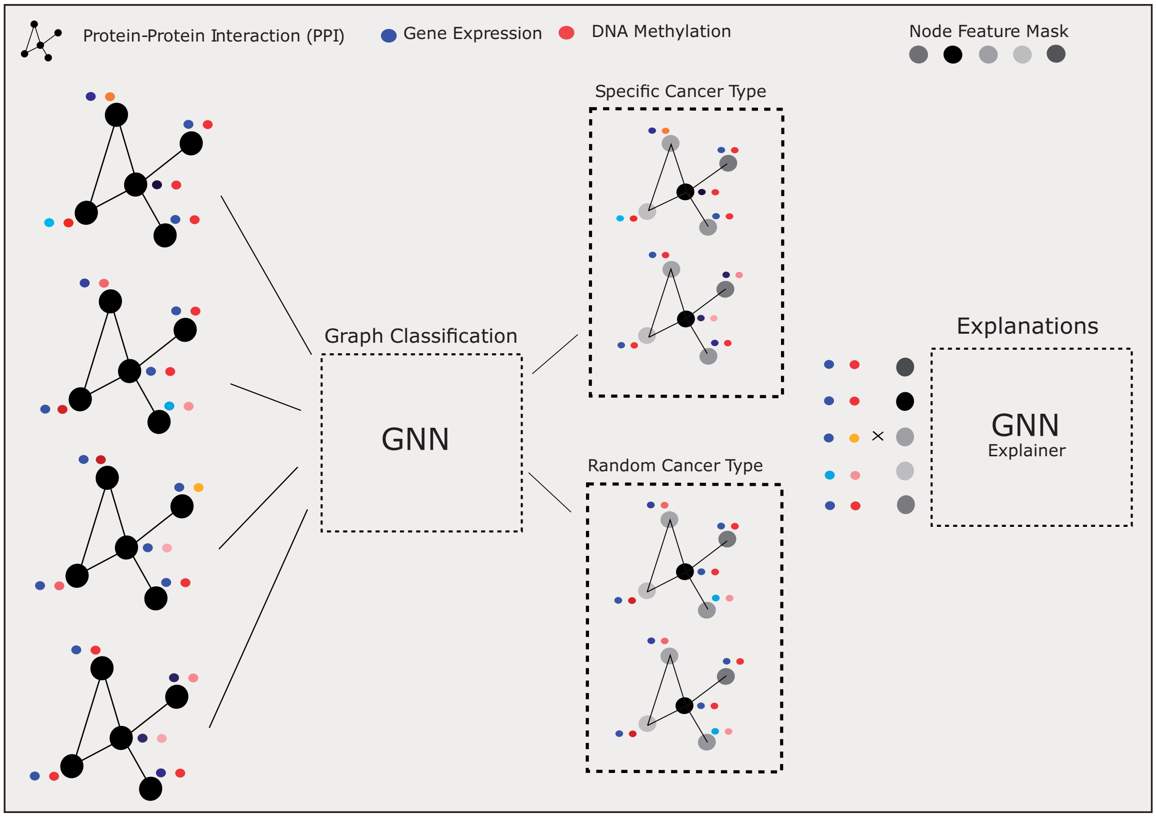

Fig. 1. Illustration of patient classification into a cancer-specific and randomized cancer group using explainable Graph Neural Networks. Each patient is represented by the topology of an protein-protein interaction network (PPI). Nodes are enriched by multi-omic features from gene expression and DNA Methylation (colored circles). The topology of each graph is the same for all patients, but the node feature values vary, reflecting the cancer-specific molecular patterns of each patient.

rule, where X_S is a subset of the features of the nodes in the subgraph G_S. $\mathbf{Y}$ represents the predicted label distribution; thereby the optimization process as a whole uses the change of the predicted label's distribution as 'guidance'.

$$\max_{G_S} \ \mathrm{MI}\left(\mathbf{Y}(G_S, X_S)\right) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{G} = G_S, \mathbf{X} = X_S) \tag{4}$$

The random variables are denoted with bold letters, whereas instantiations (possible outcomes) thereof with non-bold letters.

In our proposed framework, we employ a slightly modified version of the GNNexplainer with an induced sampling scheme. Since we apply the explanations on a graph classification task, where all graphs have the same topology, Equation (4) can be re-written as:

$$\max_{X_S} \ \mathrm{MI}\left(\mathbf{Y}(G, X_S)\right) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{G} = G, \mathbf{X} = X_S), \tag{5}$$

where G is the original graph. We employ a node mask on $\mathbf{X}$ such that a subset of nodes in X_S can be inferred to maximize mutual information with a minimal set of features.

To make the optimization process more efficient and tractable, the researchers came up with several constraints and improvements. For more details please see Ying *et al.* (2019). Accordingly, we solve the following optimizing function by gradient decent.

$$\min_{N} \sum_{c=1}^{C} \mathbb{1}[y = c]; \log P_{\Phi}(\mathbf{Y} = y|\mathbf{G}, \mathbf{X} = X \times \sigma(N)), \tag{6}$$

where $N \in \mathbb{R}^{1\times\#V}$ specifies the learned feature node mask passed through the sigmoid function σ. $X \times \sigma(N)$ is the row-wise multiplication of X, where the rows reflect the nodes and the columns are representing the features. The number of nodes in X can be constrained; this is a configurable parameter in the implementation provided in the GitHub repository. In the above expression c denotes one of the possible C classes of a classification class.

GNNexplainer allows for node feature masks as well as edge masks. The GNNexplainer, however, may not be applicable for *model-wide* explanations (Luo *et al.*, 2020). This is due to the fact that it optimizes a specified node and edge mask with regard to a single input graph. As a consequence, explanations may not reflect the *global* decisions made by the GNN classifier (Luo *et al.*, 2020). In fact, the mentioned problem was recently addressed by a method called PGExplainer (Luo *et al.*, 2020). However, the PGExplainer explicitly works on edge masks and thus requires the GNN model to internally adjust edge weights, which was not applicable in our case. Thus, we propose a slight modification of the GNNexplainer for *model-wide* explanations.

We randomly sample graphs from the input space, while optimizing one single node feature mask $N \in \mathbb{R}^{1\times\#V}$. After a certain number of epochs the sampling scheme is repeated. As a result, the values of the node feature mask converge to *global* node importance values. This approach is very much related to classical feature selection. Instead of inferring explanations for a single instance, we provide feature importance values for the whole set of samples. The proposed important node attributes may be an efficient technique for GNN-based dimension reduction, so that reduced subnetworks could provide more parsimonious models which to this end may generalize better on unseen test data.

Ultimately, disease subnetworks are detected with the following approach. First, we assigned edge relevant scores by calculating the average node feature importance of two connected nodes, inferred by our modified GNNexplainer. The obtained edge-specific scores are used to weight the edges of the PPI network. Following, a Louvain method (Blondel *et al.*, 2008) for community detection was applied to the weighted input graphs. The detected communities are

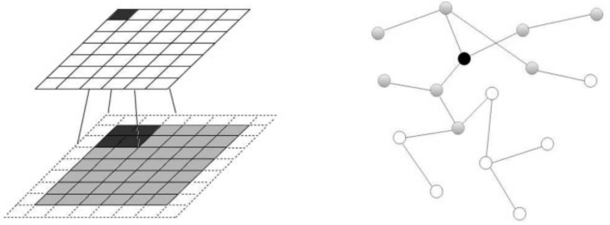

Fig. 2. On the left side of the figure, the input and output grid of a two-dimensional convolution operation with a kernel of size 3 x 3, zero padding, and stride 1 is depicted. The kernel itself is not shown in the figures. The kernel slides over parts of the input grid. On the right side, the two-hop neighborhood of the centre node (marked with black color) is painted gray. The features of the neighboring nodes will be used to define updated values for the features of the currently processed node

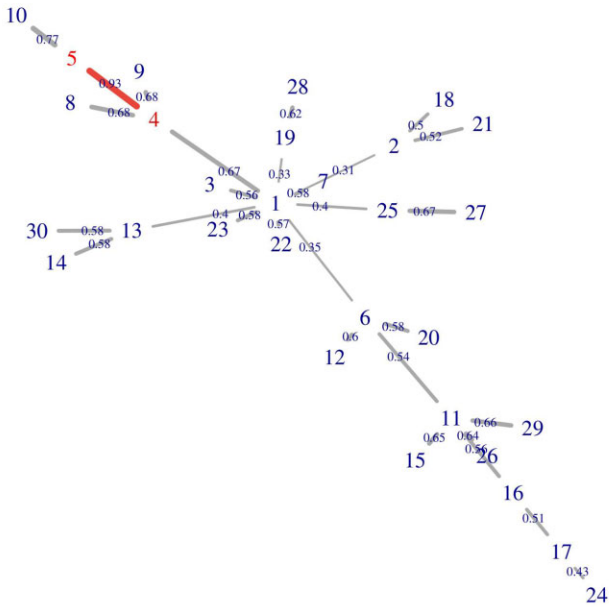

Fig. 3. Barabasi graph. Shown is a simulated Barabasi graph with 30 nodes and 29 edges. The edges are labeled by the importance scores obtained from our modified GNNexplainer. Selected edge is 4–5 with the highest score of 0.93

ranked according to their average edge importance scores. The top-ranked community represents the detected disease subnetwork.

3.3 Sanity checks on synthetic data

We have validated our approach on synthetic Barabasi networks (Fig. 3). We have generated 1000 networks which each consists of 30 nodes. Node feature values were generated using a normal distribution with $N(\mu = 0, \sigma)$. Following, we randomly sampled two connected nodes for which we assigned feature values from $N(\mu = -1, \sigma)$ for one half of the networks, and values from $N(\mu = 1, \sigma)$ for the rest. We evaluated whether and to what extend the GNNexplainer successfully uncovers the selected edge and the corresponding nodes. We varied the σ values to evaluate the stability and robustness of the explanations. Results of these sanity checks are shown in Figure 4 and Table 1 of Section 4.1.

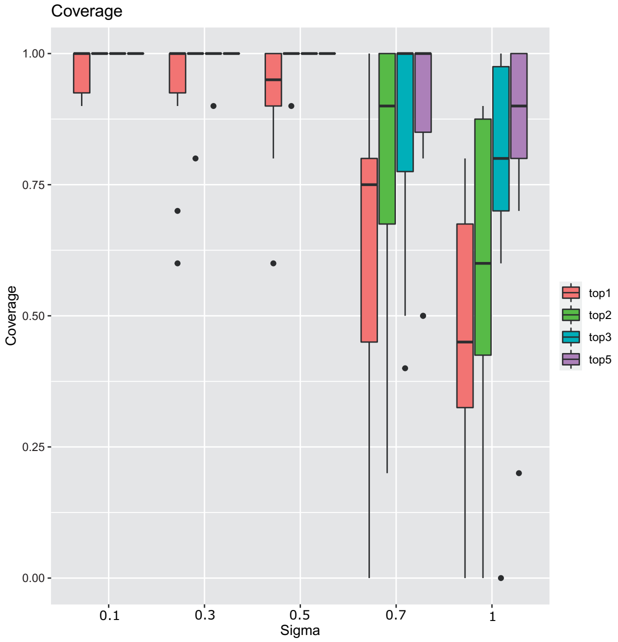

Fig. 4. Results on synthetic Barabasi graphs. Shown is the coverage, which is measured by the number of times the selected edge is ranked within the top-k elements. The edge importance values are calculated based on the GNNexplainer with induced sub-sampling to obtain one single node mask

Table 1. Performance of the GNN classifier and its explanations

Added noise σ	GNN Accuracy (%)	Coverage of GNNexplainer Model-wide (ours)/instance-level
0.1	100	1/0.5
0.3	98	1/0.5
0.5	91	0.94/0.5
0.7	79	0.75/0.4
1	70	0.44/0.3

3.4 The Cancer Genome Atlas human cancer data

We have downloaded molecular multi-modal data from the *linkedomics.org* server (Vasaikar *et al.*, 2018). The authors provide harmonized multi-omics datasets retrieved from The Cancer Genome Atlas (TCGA) database (https://www.cancer.gov/tcga), which represents one of the largest collections of multi-omics datasets. It contains molecular and genetic profiles for over 33 different cancer types from 20 000 individual tumor samples (Subramanian *et al.*, 2020). In this work, we analyzed three different cancer types, namely kidney renal clear cell carcinoma (KIRC), breast invasive carcinoma (BRCA) and lung adenocarcinoma (LUAD), for which we have detected cancer-specific subnetworks that are substantially different from a randomized control group (1). The control group consists of 200 randomly sampled patients across randomly selected cancer types.

We have downloaded gene expression (HiSeq) and DNA methylation (HM450K) to enrich the nodes with patient-specific feature values (see Fig. 1). The datasets were harmonized so that for every patient multi-source information is available. Furthermore, we only focused on cancer-relevant genes as proposed by Schulte-Sasse *et al.* (2021). Genes containing missing values at least for one patient were removed from the analyses. The obtained numerical data matrices were normalized using *min–max* normalization. The PPI network was retrieved from the STRING database (Szklarczyk *et al.*, 2021). We only kept genes for which both, mRNA gene expression and DNA methylation data were available. We deleted edges with relevance scores lower than the 95-percentile. In case this filtering resulted in a multi-graph, we kept the subnetwork with the highest number of nodes.

4 Results

4.1 Sanity checks on synthetic data

Results obtained from synthetic data indicate that our proposed GNNexplainer node feature mask successfully uncovers the GNN

Table 2. Classification accuracy (min/median/max) on TCGA cancer data

Classifier	Cancer	mRNA	DNA methylation	Multi-omics
GNN-SubNet	KIRC	53/57/76	75/83/91	79/85/91
	BRCA	49/55/65	51/71/81	69/76/85
	LUAD	49/51/59	78/87/94	79/91/95
DFNET	KIRC	75/78/84	78/84/86	79/83/89
	BRCA	62/70/78	69/70/83	73/76/83
	LUAD	80/87/96	79/83/88	80/86/91
NN	KIRC	82/86/91	82/87/92	82/86/93
	BRCA	65/74/85	71/77/82	56/78/86
	LUAD	81/87/91	83/87/92	82/88/95
RF	KIRC	80/83/90	82/87/94	81/88/92
	BRCA	65/76/86	67/79/85	71/78/82
	LUAD	77/87/92	82/89/94	85/88/94

black-box decisions. Figure 2 shows an example of a simulated Barabasi graph whose node features are generated with $N(\mu = 0, \sigma = 0.1)$. The graph consists of 30 nodes and 29 edges. We simulated 1000 graphs with the exact same topology but with varying node feature values. The feature values of the selected edges 4–5 were generated from $N(\mu = -1, \sigma = 0.1)$ for 500 graphs, and from $N(\mu = 1, \sigma = 0.1)$ for the other 500 graphs. Our sampling induced variation of the GNNexplainer for *model-wide* explanations successfully detects the selected edge with the highest score of 0.95 (see Fig. 2). Notably, the selected edge is detected even though it is not placed within a highly connected community. This observation suggests that the GNN classifier as well as the explainer are not biased towards nodes with high edge degree.

We repeated the analyses with varying graph topologies and variable σ values of the selected edge node features. As can be seen from Figure 3, the selected edge is within the top-2 ranked edges in all cases, when σ values are lower than 0.5. For $\sigma > 0.5$, the accuracy of the GNN classifier reduces significantly, and explanations get worse accordingly. For $\sigma = 0.3$ a single outlier run with low coverage could be observed. However, the median coverage values are still at 100%. Table 1 shows the median coverage values for the top-1 ranked edges as well as the accuracy of the GNN classifier. As expected, the more noise is added to the synthetic data, the lower the accuracy of the GNN classifier.

The original implementation of the GNNexplainer generates explanations for each graph independently and thus provides *instance-level* explanations, which may deviate from the model-wide signal. In Table 1, we report on this shortcoming. Our proposed *model-wide* GNNexplainer is much more accurate for this specific task. Interestingly, in the case of $\sigma = 0.3$ explanations are at 100% while the overall accuracy of the GNN model is at 98%. The high accuracy of the explanations is due to the fact that the GNNexplainer perturbates the features so that the predicted class becomes most likely. Thus, we believe our modified GNNexplainer could be useful as a feature selection algorithm, where the most important nodes are filtered and a new classifier could be trained on this reduced set. Additional experiments were conducted accordingly (see Section 4.2).

4.2 Application to TCGA cancer data

The GNN model was trained using a validation set, and we applied an early stopping criteria. Test set performance is reported based on a 80–20 train-test data split. The number of epochs was set to 10, and we kept the model with lowest *loss* value on the validation set. This model was finally applied to the hold-out test dataset. Train-test splitting was executed 20 times and we report on the min/median/max test set accuracy (see Table 2). The accuracy of the employed GNN model is shown in Table 2.

We could observe that in case of GNN-SubNet incorporating multiple biological sources was beneficial in all cases, when judged by the median accuracy. Interestingly, mRNA as a single-source node feature performed worse than DNA Methylation for all analyzed cancer types. This observation might indicate that KIRC, BRCA and LUAD have similar mRNA gene expression levels, but differ mostly due to epigenetic factors.

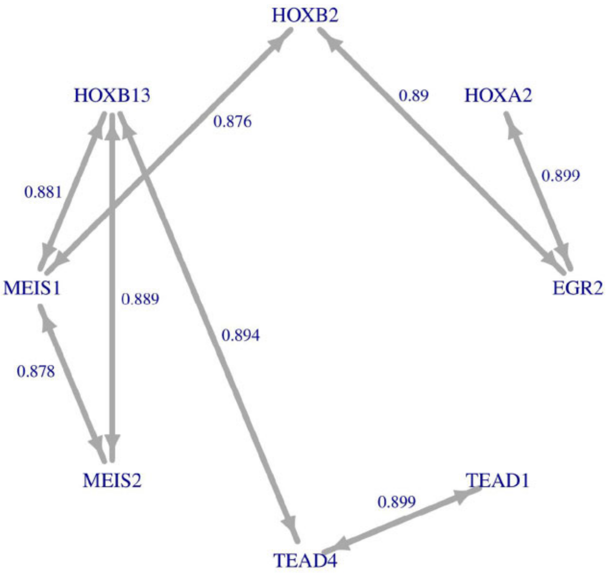

Fig. 5. Kidney cancer (KIRC) disease subnetwork. Shown is the top-ranked community inferred by our proposed explainable GNN pipeline. Edge weights are calculated using the GNNexplainer with induced sub-sampling for model-wide explanations. The top-2 and top-3 module can be found in the Supplementary Figure S3

We compared the GNN classifier with the DFNET method (Pfeifer *et al.*, 2021), which was specifically developed for disease subnetwork discovery. We initialized 100 trees on the PPI network which we then let evolve 10 greedy iterations. The *mtry* parameter, which defines the depth of the random walks for building the trees, was set to 30. The predictive accuracy was similar to the GNN classifier (Table 2), but had slightly lower accuracy for the kidney cancer (KIRC) and lung cancer (LUAD) dataset. In contrast to the GNN classifier DFNET was more accurate on single-modal gene expression data. This observation can also be made when classical machine learning techniques are applied, such as random forest (RF) and neural network (NN; Table 2). For these classifiers, we do not incorporate any domain-knowledge about the interaction and relatedness of genes. Results indicate that GNN-SubNet has similar performance to alternative approaches. This is an intriguing result since we aim for increased interpretability while maintaining predictive performance.

To illustrate disease subnetwork detection with explainable AI we conducted further analyses based on the KIRC dataset. We applied our modified GNNexplainer on the KIRC test set data, which consists of 80 patients within the test set, in order to verify the most important network regions for classification. From initially 2049 genes and 13 588 edges, we have detected 36 modules in total. The top-ranked module, according to its average edge importance score, is shown in Figure 5. The module consists of eight genes, namely EGR2, HOXA2, HOXB13, HOXB2, MEIS1, MEIS2, TEAD1 and TEAD4. These genes are connected by eight edges. The genes HOXB13 and MEIS1 have the highest connectivity with three edges, where HOXB13 has the highest average edge importance values with $[0.881, 0.889, 0.894]$. HOXB13 methylation was previously reported to positively correlate with tumor grade and microvessel invasion. The authors suggest that HOXB13 is a novel candidate tumor suppressor gene in renal cell carcinoma (RCC) and that its inactivation may play an important role in both RCC tumorigenesis and progression (Okuda *et al.*, 2006). An additional study verified the G84E mutation with the HOXB13 genes as an increased risk of prostate cancer (Hoffmann *et al.*, 2015).

Table 3. Accuracy (min/median/max) of the selected kidney cancer (KIRC) genes

Method	Selected genes	Accuracy	PPI network distance Scores: 0, 100, 300, 500, 700 and 900	GO enrichment *P*-value
GNN-SubNet	**MGAT3, MGAT4B, MGAT5** and **MGAT5B**	77/79/80	2,2,2,2,3,3	0.3^{-10}
DFNET	**HNRNPM**, SNRNP200, XAB2, **SF3B3, HNRNPUL1, CTNNBL1, CDC5L, BCAS2, HSPA8, RBM5**, SF3B4, SRRT, **CDC40** and **PRPF4**	76/80/85	2,2,2,2,2,2	0
NN	ATP2A1, **SLC17A7, BMPR1B** and CLK2	82/88/93	3,3,4,4,5,6	0.4^{-4}
RF	ANKRD11, DSTYK, **CDH8** and **FBF1**	78/87/95	3,3,4,5,5,5	0.3^{-4}

The genes with enriched GO terms are in bold.

As can be seen from Figure 5, the HOXB13 interacts with the genes MEIS1 and MEIS2. MEIS proteins are transcription factors and bind direct HOX protein activity as putative tumor suppressors (VanOpstall *et al.*, 2020). An additional transcription factor within the detected disease module is EGR2. It maintains high expression of IGF2BP proteins, which are overexpressed RCC (Ying *et al.*, 2021). The detected module and the corresponding gene interactions may serve as novel biomarker. The contribution of each single-omic to the predictions can be obtained from Supplementary Figure S4.

In a second experiment, we trained the GNN classifier on the entire KIRC dataset, including all 400 patients. GNN-SubNet verified a module consisting of four genes, namely MGAT3, MGAT4B, MGAT5 and MGAT5B. We trained a new GNN model from scratch based on these genes and calculated the test set performance based on an 80–20 train-test split. This procedure was repeated 20 times and the min/median/max performance is reported (Table 3). The contribution of each omic to the predictions can be found in Supplementary Figure S5. For the competing methods, we verified the top-4 important genes, according to the number genes within our detected module, and a new classifier on this reduced feature set was trained. The Gini index was used for RF-based feature ranking. NN-based feature importance values were calculated accommodating mean SHAP (Shapley) values. For a detailed description of this experiment see Supplementary Section S4.

In terms of model performance, the ML approaches that do not incorporate any PPI prior knowledge have selected most powerful genes. However, the selected genes spread widely across the whole PPI network and thus may not capture a robust and strong biologically signal. Also, the difference between the min and max accuracy of GNN-SubNet is very low, which might already be an indication for higher robustness. In an in-depth further investigation, we computed the shortest path for each gene pair within the selected feature set to verify the distance between them within the PPI network. Since the connectivity of the input graph is controlled by the parameter ‘combined_score’, we performed several experiments with different graphs that were created by applying a threshold on this edge feature (Table 3). The higher the ‘combined_score’ parameter the more confident one can be about the biological relationship between the genes. As expected, NN and RF are selecting features whose PPI distances are higher than for DFNET and GNN-SubNet. Some gene pairs do not have any path when high edge confidences are retained (see Supplementary Fig. S11). Finally, we performed GO enrichment analyses for which we obtained the most significant results for the gene sets selected by GNN-SubNet and DFNET (Table 3).

5 Conclusion

In this article, we have presented a novel way to the disease subnetwork detection based on PPI networks and explainable GNN. We argued that the incorporated PPI knowledge-graph restricts the deep model to learn on more reliable and biological meaningful trajectories compared with classical DL approaches. This is important and it may fasten the way towards trustworthy AI and the discovery of novel biomarker.

We have introduced a simple modification of the GNNexplainer program such that it computes model-wide explanations. This was realized by randomly sampling networks from the input space, while optimizing a single node mask. From this node mask edge, relevance scores were computed and assigned as edge weights to the PPI network. Disease subnetworks are finally inferred by a weighted community detection algorithm. We have demonstrated PPI disease subnetwork detection from patients suffering kidney cancer. Each patient was modeled as a PPI network comprising multi-omic node features from mRNA gene expression and DNA methylation data.

Finally, we have implemented our proposed methodologies within the GNN-SubNet program. We plan to develop this program further and add features from various angles. For instance, several additional GNN-based explainers will be incorporated. The GNN-LRP explainer is of particular interest. GNN-LRP explains the GNN classifier using higher-order expansions (Schnake *et al.*, 2020). A main advantage compared with other methods is that it is capable of reporting on both, positive as well as negative contribution of features to a particular prediction. This additional capability could help to increase the interpretability of the detected disease subnetworks.

Acknowledgements

We are grateful for the kind assistance of Afan Secic, software engineering student at the time of this project.

Funding

This paper was published as part of a special issue financially supported by ECCB2022. Parts of this work have been funded by the Austrian Science Fund (FWF), Project: P-32554 ‘explainable Artificial Intelligence’. Parts of this work have received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 826078 (Feature Cloud). This publication reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

Conflict of Interest: none declared.

Data availability

The TCGA cancer data underlying this article are available at http://www.linkedomics.org/, the PPI networks can be downloaded from the STRING database (https://string-db.org/). We have implemented our approach within the Python package GNN-SubNet, freely available on GitHub (https://github.com/pievos101/GNN-SubNet).

References

Bacciu,D. *et al.* (2020) A gentle introduction to deep learning for graphs. *Neural Netw.*, **129**, 203–221.

Blondel,V.D. *et al.* (2008) Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.*, **2008**, P10008.

Choobdar,S. *et al.*; DREAM Module Identification Challenge Consortium. (2019) Assessment of network module identification across complex diseases. *Nat. Methods*, **16**, 843–852.

Hoffmann,T.J. *et al.* (2015) Imputation of the rare HOXB13 G84E mutation and cancer risk in a large population-based cohort. *PLoS Genet.*, **11**, e1004930.

Holzinger,A. *et al.* (2019) Interactive machine learning: experimental evidence for the human in the algorithmic loop. *Appl. Intell.*, **49**, 2401–2414.

Holzinger,A. *et al.* (2021a) Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence. *Inf. Fusion*, **79**, 263–278.

Holzinger,A. *et al.* (2021b) Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI. *Inf. Fusion*, **71**, 28–37.

Jeanquartier,F. *et al.* (2015) Integrated web visualizations for protein-protein interaction databases. *BMC Bioinformatics*, **16**, 195.

Ji,S. *et al.* (2021) A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.*, **33**, 494–514.

Kipf,T.N. and Welling,M. (2016) Semi-supervised classification with graph convolutional networks. *arXiv*, *preprint arXiv:1609.02907*.

Koller,D. and Friedman,N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, Massachusetts, US.

Kulmanov,M. *et al.* (2020) Machine learning with biomedical ontologies. bioRxiv. https://doi.org/10.1101/2020.05.07.082164.

Liu,G. *et al.* (2009) Complex discovery from weighted PPI networks. *Bioinformatics*, **25**, 1891–1897.

Luo,D. *et al.* (2020) Parameterized explainer for graph neural network. arXiv, *preprint arXiv:2011.04573*.

Okuda,H. *et al.* (2006) Epigenetic inactivation of the candidate tumor suppressor gene hoxb13 in human renal cell carcinoma. *Oncogene*, **25**, 1733–1742.

Pearl,J. (2019) The seven tools of causal inference, with reflections on machine learning. *Commun. ACM*, **62**, 54–60.

Pfeifer,B. *et al.* (2021) Network module detection from multi-modal node features with a greedy decision Forest for actionable explainable AI. arXiv, *preprint arXiv:2108.11674*.

Saranti,A. *et al.* (2019). Insights into learning competence through probabilistic graphical models. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, Springer, Heidelberg, Germany, pp. 250–271.

Scarselli,F. *et al.* (2009) The graph neural network model. *IEEE Trans. Neural Netw.*, **20**, 61–80.

Schnake,T. *et al.* (2020) Higher-order explanations of graph neural networks via relevant walks. arXiv, *preprint arXiv:2006.03589*.

Schulte-Sasse,R. *et al.* (2021) Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. *Nat. Mach. Intell.*, **3**, 513–526.

Subramanian,I. *et al.* (2020) Multi-omics data integration, interpretation, and its application. *Bioinform. Biol. Insights*, **14**, 1177932219899051.

Szklarczyk,D. *et al.* (2021) The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.*, **49**, D605–D612.

Tiddi,I. and Schlobach,S. (2022) Knowledge graphs as tools for explainable machine learning: a survey. *Artif. Intell.*, **302**, 103627.

VanOpstall,C. *et al.* (2020) MEIS-mediated suppression of human prostate cancer growth and metastasis through HOXB13-dependent regulation of proteoglycans. *Elife*, **9**, e53600.

Vasaikar,S.V. *et al.* (2018) Linkedomics: analyzing multi-omics data within and across 32 cancer types. *Nucleic Acids Res.*, **46**, D956–D963.

Vu,M.N. and Thai,M.T. (2020) PGM-explainer: probabilistic graphical model explanations for graph neural networks. arXiv, *preprint arXiv: 2010.05788*.

Wu,Z. *et al.* (2020) A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, **32**, 4–24.

Xu,K. *et al.* (2018) How powerful are graph neural networks? arXiv, 1810.00826.

Ying,R. *et al.* (2019) GNNexplainer: generating explanations for graph neural networks. *Adv. Neural Inf. Process. Syst.*, **32**, 9240–9251.

Ying,Y. *et al.* (2021) EGR$_2$-mediated regulation of m^{6}A reader IGF$_2$BP proteins drive RCC tumorigenesis and metastasis via enhancing S$_1$PR$_3$ mRNA stabilization. *Cell Death Dis.*, **12**, 1–12.

Zhang,X.-M. *et al.* (2021) Graph neural networks and their current applications in bioinformatics. *Front. Genet.*, **12**, 690049.

Zhang,Z. *et al.* (2020) Deep learning on graphs: a survey. *IEEE Trans. Knowl. Data Eng.*, **34**, 249–270.

Zhou,J. *et al.* (2020) Graph neural networks: a review of methods and applications. *AI Open*, **1**, 57–81.

Zitnik,M. *et al.* (2018) Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, **34**, i457–i466.