# GraphGONet: a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression

Victoria Bourgeais, Farida Zehraoui, Blaise Hanczar

# GraphGONet: a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression

Victoria Bourgeais*, Farida Zehraoui, and Blaise Hanczar

Université Paris-Saclay, Univ Evry, IBISC, 91020, Évry-Courcouronnes, FR

**Abstract**

***Motivation:*** *The application of sophisticated machine learning models, including deep learning, on omics data enables the emergence of precision medicine. However, their use in clinics is limited as they are not explainable. Domain knowledge can contribute to the production of accurate and intelligible predictions. Therefore, knowledge-based deep learning models appear to be a promising solution.*

***Results:*** *In this paper, we propose GraphGONet, a new self-explaining neural network that encapsulates the Gene Ontology in its hidden layers. Each neuron in the layers represents a biological concept, combining the gene expression profile of a patient and the information from its neighbors. The experiments confirm that our model not only performs as accurately as the state-of-the-art (non-explainable ones) but also automatically produces stable and intelligible explanations composed of the most contributing biological concepts. In summary, our tool is applicable by medical experts.*

## Introduction

The objective of precision medicine is to propose medical solutions at different stages of the health care pathway (diagnosis, prognosis, treatment), considering the unique low-scale characteristics of the patients known as omic profiles. Among these characteristics, gene expression profile (GE) (i.e., transcriptomic data) is an indicator of the cellular state that can help to understand the complexity of diseases such as cancers. The analysis of this data can be achieved by automatic algorithms such as machine learning (ML) algorithms (1). These algorithms build classifiers to predict phenotypes and identify GE signatures. Deep learning (DL), a successful branch of ML especially for images and texts, starts to be applied to gene expression data and shows promising results (2; 3). It has the advantage to extract nonlinear relationships within the data through a hierarchical architecture.

One of the most challenging problems that prevents the development of ML in healthcare is the lack of interpretability of those methods. In fact, most ML algorithms, including DL approaches, are considered as black-boxes. It means that the decisions of these models are not explainable due to their complexity. Making ML algorithms interpretable is one of the most important current issues. Especially in the medical field, final users (e.g., researchers, clinicians, patients) need to understand the reasons why a phenotype has been predicted to make sure that it is based on reliable medical features rather than on irrelevant artifacts. Regardless of the model's effectiveness, this will have a major effect on their decisions and confidence towards the model. Finally, the model inspection may contribute to biological discovery by revealing interesting signatures.

There exist two general approaches for interpreting these black-boxes: post-hoc methods and self-explaining models (4; 5). In case of post-hoc interpretation, a probing method is built on top of a black-box model to explain the predictions of it. Different post-hoc methods have been developed in the literature. Among them, surrogate models, which are explainable methods, approximate black-box models. For example, considering a given prediction, *Local Interpretable Model-Agnostic Explanations* (LIME) can approximate locally any black-box model by a linear method (6). Alternatively, the self-explaining models are capable to produce their own explanations to their predictions. They can be considered as explainable as the following standard models: decision trees, rules systems, and linear sparse models. However, these three methods perform less on high-dimensional complex data compared to more sophisticated ML models such as deep learning. The most interesting work on self-explaining DL models attempts to imitate a linear model with a neural network (7). The development of self-explaining models is becoming popular as it can solve disadvantages from dissociating model prediction from model explanation. For example, using different post-hoc methods can lead to different explanations, since the approximations cannot reproduce perfectly the general behavior of the original model (8). Therefore, a part of the ML community encourages the development of high-accurate self-explaining models (8; 5). Furthermore, domain knowledge is necessary to complete the obtained explanations, the aim is at making them intelligible to final users (9).

In precision medicine, the knowledge gathers different types, including pathways (KEGG (10), Reactome (11)), functions (Gene Ontology (12)), networks of interactions (STRING (13)), etc.

*Correspondence: victoria.bourgeais@universite-paris-saclay.fr

As far as we know, the knowledge is mostly structured as graphs and can be integrated into knowledge-based methods. The two mainstream methods contain feedforward Neural Network (NN) and Graph Neural Network (GNN). The former integrates the knowledge as a constraint on its architecture, limiting the full expressiveness of the semantic of the knowledge. The latter handles graph data, but it is not optimized to deal with some particular types of graphs, such as the directed acyclic graphs (DAGs). However, most existing works related to these two approaches are not self-explaining.

We propose in this paper GraphGONet, a self-explaining NN based on gene expression data encapsulating the Gene Ontology. The Gene Ontology (GO) has the advantage to provide information about the biological processes implied by the genes. GraphGONet is a new architecture combining the advantages of the GNN for the GO knowledge integration and the feedforward NN for the propagation of gene expression and the prediction. In addition to accurate predictions, our model is able to produce automatic explanations as the last layer of the NN contains the set of the most important neurons for the prediction and their associated GO terms in such way that the outcome of a patient is directly explained by this set. An enrichment test is therefore no longer required, making GraphGONet directly usable by clinicians.

The paper is organized as follows. First, we review the literature on knowledge-based methods using gene expression data for precision medicine. Then, the architecture and the learning procedure of GraphGONet are presented. Finally, we assess the model effectiveness for cancer detection on two publicly available datasets and compare it with the state-of-the-art ML methods. Case studies are provided to show how to get an explanation of an outcome and an interpretation of the model.

## Related works

In the context of DL for precision medicine from gene expression data, two knowledge-based approaches have been used to integrate knowledge into the model: the feedforward NN with constrained architecture and the Graph Neural Network. In both cases, the biological concepts are associated to neurons, understandable by the users.

The first approach, sometimes referred as Visible Neural Networks in the bioinformatics literature (14), is based on feedforward NNs. The architecture of the network (e.g., multilayer perceptron (MLP)) mimics the architecture of the knowledge graph. Each neuron in the hidden layers corresponds to a biological concept, and each connection between two neurons to a relation within the graph. The propagation of the gene expression through the network is limited to the connections corresponding to the edges of the knowledge graph. For example, Kang et al. use the gene regulator graph to connect genes (of the input layer) to the neurons (of a hidden layer) representing proteins or compounds regulating the genes (15). In Deep GONet, each hidden layer of a fully connected NN represents a level of GO (16). The association neuron-GO term is achieved during the training by penalizing the connections that do not correspond to edges in GO. In P-NET, the knowledge from Reactome allows defining a network with gene expression, methylation, mutation and copy number input data propagated to three hidden layers representing respectively genes, pathways and biological processes (17).

There exist some limitations to this approach. The integration of the knowledge is restricted by the type of the model architecture. By definition, a feedforward NN can represent only DAGs that do not correspond to some knowledge graphs (KEGG, STRING...). Moreover, the input genes are only connected with the first hidden layer and not with the deeper layers. The connections between non-successive layers are also omitted. A part of the knowledge must therefore be truncated to be integrated into the neural network.

The second approach consists in adapting existing methods to deal directly with different types of graph (directed or undirected). It can help to overcome the above limitations. It only depends on the capacity of the methods to handle this type of non-Euclidean data. A recent type of DL model, i.e., Graph Neural Networks (GNN), has been proposed to process graphs (18). The objective of these models is to propagate recursively the information contained in the nodes to their neighborhood in order to solve the prediction task (19). They can be utilized for node, edge, and graph prediction. They have been already used in biological applications to predict the label of molecules, atoms, or bonds (20). To the best of our knowledge, few works have been published for phenotype prediction on GE. Phenotype prediction is a graph-level based problem where a patient is represented as a graph whose nodes contain the information about the patient expression profile. Most works use undirected gene interactions graphs such as protein-protein interaction (PPI) networks from STRING database (21) or co-expression graphs (22) to define their input layer. The data are then propagated to some graph convolutional and pooling layers, and finally fully connected and output layers. These approaches are mainly concerned with the maximization of the accuracy and outperforming the state-of-the-art. They base their work on primer GNNs that are not self-explaining and not easily interpretable (23; 24). Some of them try to inspect the model in a post-analysis to make it interpretable. They generally adapt the post-hoc methods to deal with this type of NN, and perform

enrichment statistical tests to identify the underlying biology expressed at the disease and model level.

Our proposed method is based on a new architecture that tackes advantages from GNNs and feedforward NNs to deal with the limitations mentioned above. The knowledge graph (here GO) is completely integrated into a self-explaining model that produces accurate predictions.

## Methods

In the following, we present GraphGONet, a self-explaining NN, whose input layer corresponds to the gene expression profile of a patient, and its hidden layers integrate knowledge from Gene Ontology. The method is illustrated in Figure 1 and shows that the signal starting from the gene input layer is propagated sequentially through the GO layers. Then, the signal passes by a selection layer, where it is concatenated and masked to achieve the prediction task in the output layer. A full description of the method is provided in the following subsection.

### The architecture of GraphGONet

Let $(X, Y)$ be a training example, where $X = [x_1, ..., x_d]$ is the gene expression profile of a patient with $d$ the number of genes, and $Y = \{0,1\}^C$ is the indicator of its class that we want to predict with $C$ the number of classes. $y_c = 1$ when the sample belongs to the class $c$, and $y_c = 0$ otherwise. Note that each sample only belongs to one class. A neuron in the input layer receives the expression of one gene. The input layer is connected to a set of neurons organized in layers, which mimics the architecture of GO. Each layer in the hierarchy represents a GO level where the first hidden layer corresponds to the most specific level (layer one in Fig. 1) and the last hidden layer represents the root (layer six in Fig. 1). Each neuron in these layers represents a GO term, and each connection between two neurons represents a relation between two GO terms. The connections are oriented from lower GO levels to upper GO levels. Note that two neurons, representing non-related GO terms, are not connected. There also exist connections skipping some layers since GO terms of non-adjacent levels can be linked (e.g., the relation between GO:0044283 and GO:0044281 in Fig. 1). Let $G(v)$ be the set of genes associated to a GO term corresponding to a neuron $v$ in GraphGONet and $\mathcal{N}(v)$ the set of neurons corresponding to the children of the neuron $v$. The gene expression is propagated to neurons through connections representing relations between genes and GO terms. The neuron $v$ is only connected to the genes in $G(v)$ and is not connected to the other genes. The activation value of a neuron $h_v$ is computed from both the expression vector $X_{G(v)}$ restricted to the genes in $G(v)$ and the activation of its child neurons in $\mathcal{N}(v)$. The activation $h_v$ of the neuron $v$ is defined as follows:

$$h_v = \begin{cases} \sigma\left(w_G h_{G(v)} + w_{\mathcal{N}} h_{\mathcal{N}(v)}\right) & \text{if } |\mathcal{N}(v)| > 0 \\ \sigma\left(h_{G(v)}\right) & \text{if } |\mathcal{N}(v)| = 0 \end{cases} \tag{1}$$

where $w_G, w_{\mathcal{N}}$ are parameters to learn, $|\cdot|$ is the cardinality, $\sigma$ is the tanh activation function ($\tanh(x) = (\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)})$). In our model, the choice of the tanh function is more relevant than ReLU. The tanh function will saturate the neurons selected in the next part of the network (the selection layer), to values close to +1 or -1, making the interpretation of the prediction much easier. $h_{G(v)}$ and $h_{\mathcal{N}(v)}$ correspond to the embedding of respectively the expression of the gene set $G(v)$ and the activation of the neurons set $\mathcal{N}(v)$, given by:

$$h_{\mathcal{N}(v)} = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} h_u \tag{2}$$

$$h_{G(v)} = W_v X_{G(v)} + b_v \tag{3}$$

where $(W_v, b_v)$ are parameters to be learned.

The activation of the neurons must be computed sequentially from the most specific neurons to the root (in Fig. 1, from GO:0019319 to GO:0000152). This sequential processing is important since the activation of a neuron depends on the activation of its children. However, the neurons having their neighborhood information available at the same time are processed simultaneously. Note that the parameters $(W_v, b_v)$ of the connections between a neuron and its associated genes are specific, whereas the parameters $(w_G, w_{\mathcal{N}})$ propagating activations through the GO layers are common to all neurons. Compared to the feedforward NN, this sharing of parameters, that is inspired from GNNs, reduces strongly the number of parameters to learn.

The next part of the model is the selection of the most activated neurons in absolute value. Their associated GO terms will be used as the support of the explanation of a prediction. The process consists of (1) concatenating the activation of all neurons, except those of the input layer, such as $H_{concat} = \text{CONCAT}(h_v | \forall v)$, (2) computing a mask $M$ identifying the most activated neurons $M_v = 1$ if $v \in \text{top}(r)$, $M_v = 0$ otherwise, where $r$ is the selection ratio and top is a function returning the indices of the $\lceil nr \rceil$ neurons selected, (3) applying the mask to select the neurons $H_{select} = H_{concat} \cdot M$. Note that $r$ is a hyperparameter of the model to fine-tune during the training phase.

The last layer returns the output, where each neuron represents one of the $C$ classes. It is a linear combination of the neurons in $H_{select}$. The activation of the output is computed from $z_c = \sum_{j=1}^{K} h_{select,j} w_{jc} + b_c$, where

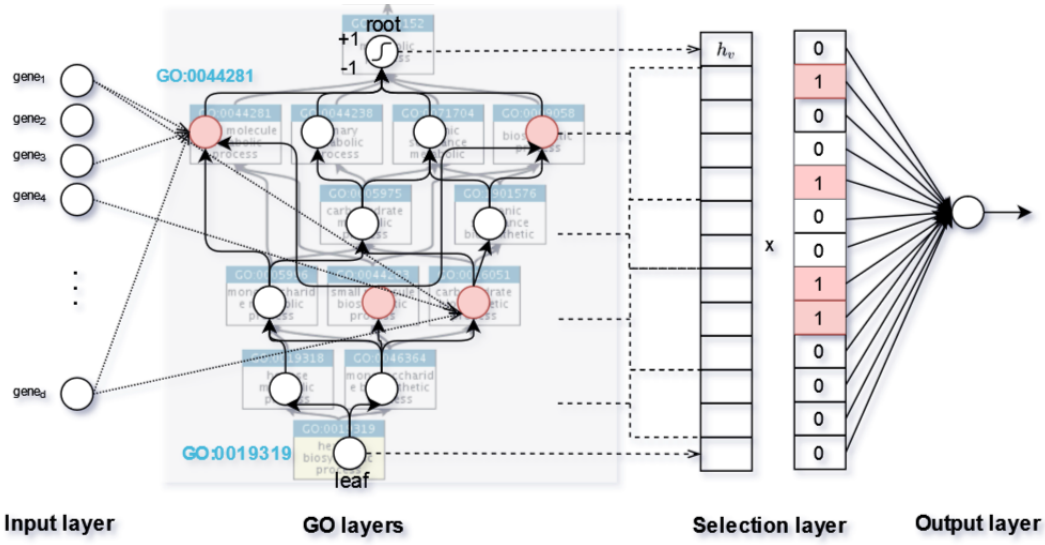


**Figure 1:** *Illustration of GraphGONet. The neurons in the input layer receive the signal from the genes. The dotted arrows correspond to the connections between the genes and the GO terms represented by neurons in the hidden layers. The relations between GO terms are represented by the plain arrows. The dashed arrows depict the concatenation of the activation of the neurons. The selection layer results from the concatenation and masking operations.*

$(w_{jc}, b_c)$'s are the parameters to learn in this last layer and $K$ the dimension of $H_{select}$ that corresponds to the number of GO terms. The output activations are transformed into probabilities using the softmax function: $o_c = \frac{\exp(z_c)}{\sum_{j=1}^{C} \exp(z_j)}$. Note that for a binary classification problem, the output layer may contain only one neuron ($C = 1$) using a sigmoid function: $o = \frac{1}{1-\exp(z_1)}$. The probability of the positive class will be returned.

The model is trained in end-to-end with the usual gradient descent searching the parameters $W_v, b_v, w_G, w_{\mathcal{N}}, W, b$ to minimize the following cross-entropy:

$$\mathcal{L} = \sum_{c=1}^{C} (-y_c \log o_c) \tag{4}$$

It is interesting to note that GraphGONet is composed of a particular type of feedforward NN and a special case of GNN. Indeed, in the feedforward NN, the input layer can be connected to each hidden layer and all the hidden layers are connected to the output layer through a selection layer. The propagation of the signal through the hidden layers, that represent GO, is inspired from the propagation rules in GNNs.

## Model interpretation

For a given patient, our model provides automatically both a prediction and an explanation. The explanation takes the form of a list of GO terms implied in the final computation of the prediction, with their score of importance. The number of GO terms in the list depends on the selection ratio $r$. It is not the same set of GO terms that will be selected for each patient. For example, the four GO terms (GO:0044281, GO:0009058, GO:0044283, GO:0016051) are chosen in Fig 1. The importance of a GO term is dependent on the weight of the connection between its associated neuron in the selection layer and the output layer. Therefore, we use an interpretation metric, the relevance score, computing the proportion of the output signal passing through the neurons in $H_{select}$ and their outcoming connections. The relevance score $R_j^c$ of a GO term $j$ is calculated as follows:

$$R_j^c = h_{select,j} \times w_{jc} \tag{5}$$

Note that the GO terms not implied in the final prediction will have their score set to 0 since their activation is nullified by the mask.

# Results

## Datasets and choice of the GO layers

We apply our model on two large public gene expression datasets for cancer diagnosis. The first dataset is a result of a cross-experimental study on heterogeneous microarray data from around 40,000 Affymetrix HG-U133Plus2 chip arrays (25). It is accessible on the ArrayExpress database under the identifier E-MTAB-3732. After quality control and normalization, the dataset comprised 54,675 input probes for a total of 27,887 cancer and noncancer samples from seventeen different types of tissue. 80% of the data form the training set and 20% the test set respecting the original proportions (66% cancer, 34% noncancer). 20% of the training set is used as a validation set for the early stopping. The second set of data comes from the RNA-Seq repository in

The Cancer Genome Atlas (TCGA) plateform (26). It includes 5,982 cancer samples of 11 cancer types and 482 noncancer samples from various tissues for 56,602 input genes. They are standardized and split in the same way as the first dataset. The full description of the two datasets can be found respectively in Supplementary Tables S1 and S2.

GO is divided into three subontologies: biological process (GO-BP), molecular function (GO-MF), and cellular component (GO-CC). In these experiments, we integrate only GO-BP in GraphGONet. The GO-BP was chosen since it is the subontology often preferred by the biologists for the explanation of the predictions. However, it is possible to replace it by GO-MF or GO-CC. The details about the preprocessing of the graph can be found in the Supplementary.

## Sensitivity analysis

We conduct two experiments to assess the effectiveness of GraphGONet over the state-of-the-art. GraphGONet is trained in end-to-end manner using the optimizer Adam with an adaptive learning rate of 0.001 and a batch size of 64. Early stopping is employed with a patience of 5 and a delta of 0.001. We perform on the microarray dataset, a binary classification with a sigmoid function in the output layer and on the pan-cancer dataset, a multi-classification with a softmax function. The accuracy reported in the figures is estimated from the test set. All the experiments described below have been executed on a GPU RTX 2080Ti using PyTorch 1.7.1 and PyTorch Geometric 1.6.3.

In the first experiment, we carry on an analysis of the selection layer to measure its role in GraphGONet. This layer is a key module to make the model self-explaining. It extracts a subset of the most informative neurons and their associated GO terms to predict the final outcome. This subset can be in fact directly used as the support of the explanation of the prediction. The size of this subset is determined by the selection ratio $r$. As described in the Methods section, the choice of the GO terms realized by the selection layer in GraphGONet is based on the absolute value of the activation of their associated neurons. In this way, we evaluate the selection process and the value of the hyperparameter $r$ and compare this process with a random selection. In addition, we vary the value of $r$ in a range from 0.00005 to 1, which influences the number of selected GO terms. When $r = 1$, all the GO terms are selected. Ten models are learned for each value of $r$ with different initialization of the weights and bias.

The average and the standard deviation of the models' accuracy are reported according to the value of $r$ in Figure 2a for the microarray dataset (resp. in Supplementary Figure Sa for the TCGA dataset). We first see that in general, random selection is less performant than

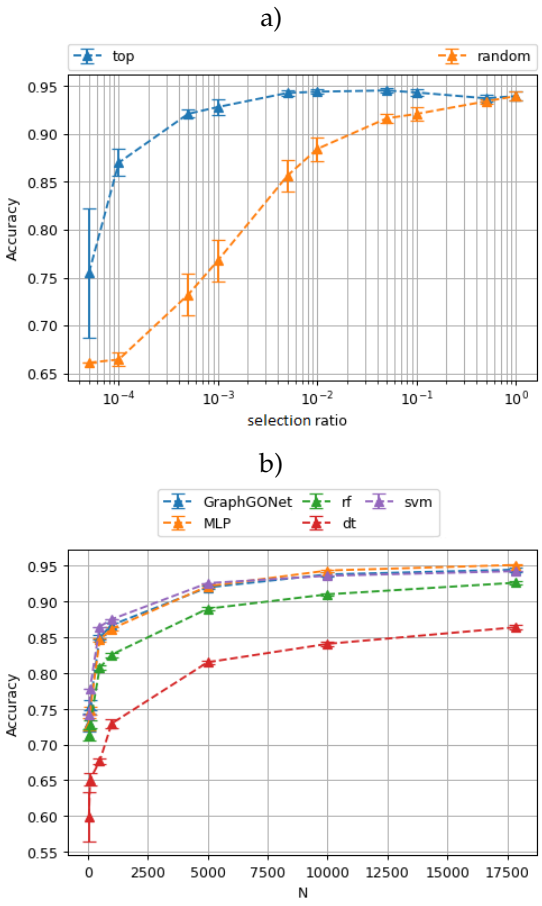


**Figure 2:** *Accuracy of the models according to (a) the selection ratio r and (b) the number of samples N on the microarray dataset.*

"top" selection on the two datasets. On the microarray dataset, performances with random selection start to decrease gradually from 0.934 at $r = 0.5$ to reach an accuracy of 0.661 at $r = 0.00005$, which corresponds to the proportion of the majority class. In contrast with "top" selection, performances increase from 0.940 to 0.945 with selection values from 1 to 0.05, and then, decrease slightly from 0.945 to 0.921 along $r$ between 0.05 and 0.0005. Applying "top" selection can help not only to interpret, but also boost the performances. By keeping all or half the neurons, performances are less good than a selection with smaller ratios. Performances finally drop to reach 0.755 at $r = 0.00005$. On the TCGA dataset, similar tendencies can be observed, except that random selection with ratios from 0.5 to 0.01 do not have much impact on the performances. It can be explained by the fact that TCGA data are more homogenous, and the task is less complicated to solve. Finding signatures proper to cancer types is easier than finding global signatures for cancer from different tissue types. Best performances are achieved with "top" selection, with a ratio of 0.05 on the microarray dataset and 0.1 on the TCGA dataset. It is interesting to note that the best performances are obtained if the prediction is based only

on a small proportion of the neurons (around 500 for microarray and 1000 for TCGA). The corresponding GO terms should therefore be related to the predicted phenotype. However, hundreds of GO terms are difficult to ingest for a human to understand the explanation. The smaller a subset of the selected GO terms is, the more understandable the explanation will be. Unfortunately, we see that after this optimal point, the model accuracy decreases with $r$. The ratio controls therefore a trade-off between performance and interpretability. This hyperparameter is still adjustable according to the user expectations.

In the next experiments, we consider models learned from two trade-offs. The first one is the best performing model with $r = 0.01$, i.e., around 100 GO terms are selected. Note that $r = 0.01$ does not correspond exactly to the ratio with the best average performances, but the difference of performances between $r = 0.01$ and $r = 0.05$ for microarray (resp. $r = 0.1$ for TCGA) is negligible, as it is less than 0.005. In the second case, we do a reasonable trade-off between performances and interpretability choosing $r = 0.001$. The accuracy decreases slightly from around 1.6% for a drop in the number of selected GO terms to around 10.

In a second experiment, we compare one of the proposed models (at $r = 0.01$) with state-of-the-art classical ML algorithms. The ML algorithms computed with the Python package scikit-learn are the following: decision tree (Gini criterion), Random Forest (Gini criterion, number of trees=100), SVM (linear kernel, C=1.0), and MLP (three layers with respectively 1000, 500 and 200 neurons). The methods are trained on different sizes of the training set in the intervals: 17847 (full size of the training set) to 50 samples for the microarray dataset, and from 4136 to 25 samples for the TCGA dataset. As for the previous experiment, ten models are learned for each sample size. Figure 2b (resp. Supplementary Figure Sb) plots the average and the standard deviation of the accuracy of each method according to the number of training samples from the microarray (resp. TCGA) dataset. We note that best accuracies are achieved with the highest number of samples, and the curves of DL methods and svm are mixed up for both datasets. Besides, all performances of the models reduce with fewer training samples. Regardless of the size of the training set, GraphGONet is as competitive as the non-explainable ML and DL algorithms and clearly outperforms the only comparable explainable method (decision tree).

## Biological analysis

In this section, we show how to propose relevant biological interpretations of the model GraphGONet and its predictions. We provide two levels of interpretation: the individual prediction level and the model level.

**Interpretation of a patient outcome**

In this subsection, we show how to provide an explanation of the predicted outcome of one patient computed by GraphGONet. The goal is to propose a predictive and transparent tool to the final users (biologists, clinicians...), which produces clear and comprehensible knowledge-based explanations by highlighting the set of the GO terms the most involved in the decision-making with their quantitative contribution. In the following, we use a selection ratio of 0.001 that leads to a model using only eleven neurons and their associated GO terms for a given prediction. We recall that the prediction of each patient will be based on different subsets of eleven GO terms. The relevance score of each GO term is computed to distinguish among the subset the most influential GO terms. In case of a softmax output, the higher the score is, the more the GO term has a positive impact on the final prediction. Regarding a sigmoid output in microarray, the sign of the relevance must be interpreted considering the cancer or noncancer outcome. The GO terms, which contribute to the cancer prediction, have a relevance score with a positive sign. If the sign is negative, the GO terms target the noncancer prediction.

In Figures 3a and 3b, we illustrate the application of our tool on a patient, from the microarray test set, correctly predicted cancer with a probability of 1 and a patient correctly predicted noncancer with a probability of 0.996. The eleven remained GO terms are reported with their relevance score according to a descending (ascending) order for the cancer (noncancer) patient. In case of the cancer patient, all the GO terms have a positive sign. Ten of the eleven GO terms are important for the prediction, as they have a score close to the average relevance score of 0.92. Only the GO:0001916 is less significant with a score of 0.14. Among the most important GO terms, we can identify some that may play a role in cancer. For example, GO:0006915 (top-1st term) and GO:0043065 (top-5th term) are related to apoptosis, which can highlight the immortality of tumor cells (27). We can observe that the terms GO:0000122 and GO:0001916 are common to the explanations of the cancer and noncancer patients. For the GO:0000122, the relevance is positive for the cancer patient and negative for the noncancer patient, but they have the same absolute contribution of 0.86 for both predictions. The rank of this GO term is slightly different between the two samples: 7 in Fig. 3a and 4 in Fig. 3b, but it remains significant for both predictions. It means that a GO term can be important for the two outcomes. The positive (negative) signal determines the prediction towards the cancer (noncancer) outcome. In contrast, GO:0001916 has a positive sign in both cases. It belongs to the set of the three terms in the explanation of the noncancer patient, which do not encode for the noncancer predic-

tion. Therefore, the relevance score helps to discern the effective impact of the GO terms on the final prediction and quantify the uncertainty of the prediction. Similar results can be achieved on the TCGA dataset. A comparative study of the explanations from a predicted BRCA (Figure Sa) and LGG (Figure Sb) patient is provided in the Supplementary.

**Interpretation of the model**

In this subsection, we give a global interpretation of the model with a selection ratio of 0.01 based on both the relevance score and the frequency of the GO terms. We first propose to measure the similarity of the explanations between the patients. We analyze the clustering of the test samples predicted cancer according to their relevance profiles. The relevance profile of a patient is based on the relevance score of all neurons from the GO layers. Relevance matrices of size $(N, K)$, where $N$ is the number of samples and $K$ the number of GO terms, are collected on the test set of each dataset. A row corresponds to the relevance profile of a patient. We apply hierarchical clustering on these matrices, using the average as the linkage criteria and the Euclidean distance as the metric. The dendrogram on the microarray dataset is plotted in Figure 4. The results on the TCGA dataset can be found in Supplementary Figure S. The first colored row below the dendrogram indicates the type of tissue of each sample (see Table S3 for details). The second row enables to distinguish the true prediction (colored in blue) from the incorrect prediction (colored in white). We can clearly discern clusters that group patients from the same tissues, especially for bone narrow (colored in orange), blood (colored in red), and lymph node (colored in cyan). Despite the fact that the model isn't designed to predict the cancer tissue type, certain neurons and their corresponding GO terms are able to extract cancer features exclusive to a particular type of tissue. One explanation can be that our model does not identify a unique cancer signature, but multiple signatures associated to the different tissues. As a deep learning model produces many splits of decision boundaries, GraphGONet succeeds to identify some signatures naturally correlated to tissue types. We can finally note that the errors are spread across clusters. However, a large part belongs to the cluster related to the blood tissue, it could be interesting to investigate the reasons why these samples are misclassified. On the TCGA dataset, each cluster formed matches perfectly a cancer type, due to the fact that the goal of the model is to predict the cancer type. The model guided by the biological knowledge is capable of projecting the data into a latent space where the classes are well separable.

To evaluate the consistency of the biological signatures, we train one hundred GraphGONet models with the same selection ratio of 0.01. Similarly to the previous result, we apply the models to the test data and compute a relevance matrix and an occurrence matrix. The dimension of these matrices is $(S, N, K)$ where $S$ corresponds to the number of models. The occurrence matrix is a boolean matrix indicating if a GO term has been selected or not by the selection layer. We can then sum up across the model axis and the patient axis the number of times a GO term is selected, resulting in a vector of size $K$. Figures Sa and Sb in the Supplementary show respectively that 40.34% of the GO terms for microarray and 62.79% for TCGA have never kept. On the opposite, some GO terms are oftentimes selected by the selection module of most models. They must contain relevant biological information for the predictions. It is the case of GO:0045944, the most frequent GO term, that appears 403K times in the experiences on microarray and 88,3K on TCGA.

To get an interpretation of the model considering the classes, we filter these matrices according to the label of each patient, and rank the GO terms based on their occurrence. We count the positive or negative sign of the relevance of the GO terms across the two first dimensions. It will indicate if the GO term is used towards the prediction in case of a positive sign or against the prediction otherwise. In Figure 5, we show an example of the top-10 most frequent GO terms on the TCGA dataset for the cancer BRCA. We see that these GO terms have an occurrence close to the upper bound of 22100, (case they are selected for each patient by all the models). For example, GO:0000122, the top-1 GO-BP term, comes out 15307. Moreover, this histogram highlights that in most cases, the more frequent a GO term is, the more the GO term is used to predict the target label. We can identify particular cases through the GO terms GO:0006614 and GO:0006958. They are used in both directions, but the mean of their absolute relevance reveals that they do not contribute as much in the computation of the prediction contrary to the other top GO terms. Comparable results can be generated for each cancer type. We observe that some GO terms are important for all the cancer types such as GO:0045944 whereas others are specific to some cancer types. On the microarray dataset, the results in Supplementary Figures Sa and Sb show again that GO terms can be used for the two predictions, but the relevance score will be positive for the prediction cancer and negative for the noncancer one.

## Discussion and Conclusion

The experiments show that GraphGONet can leverage from the whole knowledge graph (and its semantic) to accurately achieve the prediction task. We bring novelty in the sequential propagation and the selection layer, which permit to combine gene expression in an end-

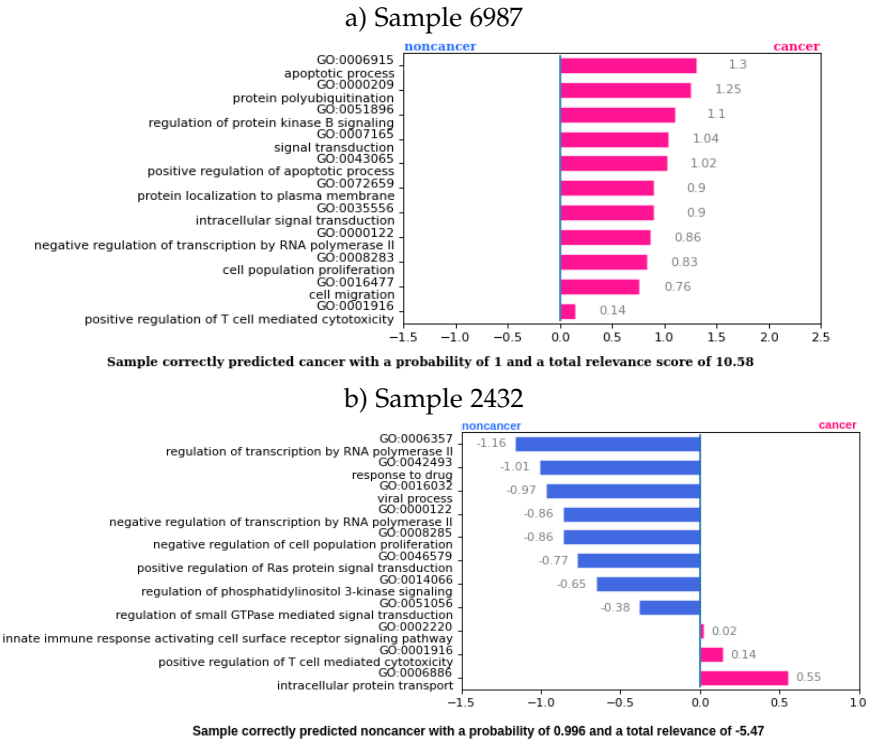


**Figure 3:** *Explanation of (a) a cancer and (b) a noncancer prediction. A subset of eleven GO terms is reported with their relevance score and their description. The color indicates towards which class a GO term influences the signal: blue for noncancer and pink for cancer. The total relevance score is the sum of the relevance scores and the bias of the output class.*

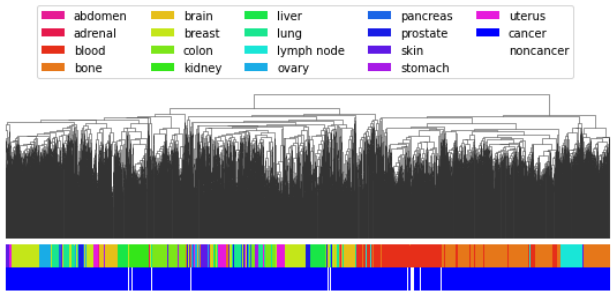


**Figure 4:** *Dendrogram of the relevance matrix on the test cancer samples from the microarray dataset. The first row displays the type of tissue of each sample, whereas the second row indicates the predicted class.*

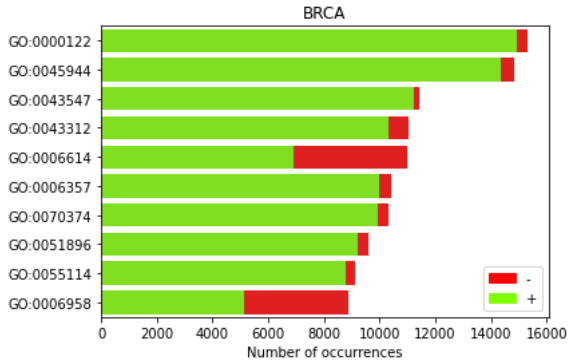


**Figure 5:** *Top-10 most frequent GO terms sorted according to their occurrence for the BRCA cancer type from TCGA. The colors indicate the part of the occurrences having a negative (red) or positive (green) relevance score. The maximal frequency that can be reached corresponds to the number of BRCA samples times the number of models, i.e., 22100.*

to-end learning and provide biological insights on the decision-makings. The propagation in the GO layers (Eq. 1) inspired by the propagation process from the graph convolutional layers enables to consider all the levels of GO and any types of connection between the neurons (adjacent, skipping). With fully connected layers, we have to cut off parts of this information because it cannot fit in memory due to a huge number of parameters to compute, besides, skipping connections are not represented. On the microarray dataset, with GraphGONet, only 291,859 number of parameters must be computed regardless of 315,766 with standard feedforward networks. The model has been validated on two datasets and can handle any knowledge represented by a DAG.

The main interest of our method is to provide easily an understandable explanation of the predictions in the form of a small set of GO terms with their associated relevance scores. However, the NNs from the state-of-the-art generally use complex and uncertain post-hoc methods to estimate the relevance of each gene. It results in a large set of relevance values, making the

explanations less understandable. The advantages of GraphGONet are that (1) the explanation is based on a subset of higher semantic concepts (GO terms instead of genes) where the subset contains few GO terms thanks to the selection layer, making the interpretation stable, (2) the relevance score is easy to compute and represents a good indicator to quantify the final contribution of the GO terms in the subset and (3) it is as accurate as explainable and different levels of explainability can be proposed depending on the user's expectations.

Specifically, for the first point, stability is one of a prerequisite of trustworthy interpretability. The use of post-hoc methods show that in some cases, a non self-explaining NN trains on the same training set can lead to different explanations due to different model parameters initialization (5). However, experiments show that our self-explaining model is stable, since it is able to extract consistent biological patterns with different initialization of the model parameters. It is worth to remind that several works have shown that the identification of gene signatures obtained from model interpretation or feature selection methods are very unstable (28). Signatures based on GO terms seem therefore more reliable. As shown in the example of the prediction of a cancer patient, these GO terms are relevant with the studied phenotype. The reliability of the system can be verified as the existence of incoherence is detectable by our model. Note that the set of genes the most important is still computable through post-hoc methods.

For the second point, the relevance score of the GO terms can be easily computed in our model (see Eq. 5). In practice, they simply depend on the weights of their associated neurons with the output layer and the sign of the activation of the neurons. The ease comes from the fact that a GO term, represented by one neuron, is directly connected to the output layer, thanks to the GO layers propagating the gene expression. Moreover, we insist on the importance of the sign of the relevance score to indicate in which way the most important GO terms impact the final outcome towards the good or wrong prediction. A change in the sign of the signal can move the predicted outcome from one class to another. It could be interesting to inspect the reasons of this change and understand the meaning of the sign, for example, if the GO term is excited or inhibited. We can also use the following information to constraint the model to find only the GO terms that go in the correct direction. When there are some GO terms that go towards the wrong prediction, it means that the model is uncertain on the signal flowing through these GO terms. We can quantify this uncertainty by dividing their cumulated scores by the total relevance score to get an estimated proportion.

Finally, for the third point, our sensitive analysis shows that at the same time to be explainable, GraphGONet is as competitive as the other ML methods. As discussed in the ablation study on the selection ratio, the focus on interpretation rather than performance depends on the final user's requirements. An explanation produced with one hundred GO terms can still be understandable for experts. It is possible to recreate the sub-graph associated to a patient and identify the most relevant parts. Depending on the final user, different types of explanation can be provided: a doctor or a patient can be more interested in the model interpretation of a patient whereas a biologist and a statistician in the global interpretation of the model.

To sum up, our proposed model show several advantages over the state-of-the-art and in addition, it fills the following desire data mentioned in (9): exploitation of the domain knowledge, validity (reproducibility), stability, reliability, quantification of the uncertainty. It makes it usable by the medical experts. We plan to include other ontologies, such as the pathways, to enrich the biological explanations and investigate the uncertainties to rectify the model.

**Availability of data and materials**

The microarray dataset is accessible from the ArrayExpress database under the identifier E-MTAB-3732. The TCGA datasets can be downloaded from the Genomic Data Commons (GDC) data portal.
GraphGONet is freely available at `https://forge.ibisc.univ-evry.fr/vbourgeais/GraphGONet.git`.

**Acknowledgements**

This article has been accepted for publication in Bioinformatics ©: 2022 Published by Oxford University Press 10.1093/bioinformatics/btac147. All rights reserved.

## References

[1] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Comput. Struct. Biotechnol. J*, vol. 13, pp. 8–17, 2015.

[2] Z. Zhang, Y. Zhao, X. Liao, W. Shi, K. Li, Q. Zou, and S. Peng, "Deep learning in omics: a survey and guideline," *Brief Funct Genomics*, vol. 18, no. 1, pp. 41–57, 2019.

[3] B. Hanczar, F. Zehraoui, T. Issa, and M. Arles, "Biological interpretation of deep neural network for phenotype prediction based on gene expression," *BMC Bioinform.*, vol. 21, no. 1, 2020.

[4] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence

(XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inform Fusion*, vol. 58, pp. 82–115, 2020.

[5] D. C. Elton, "Self-explaining AI as an alternative to interpretable AI," in *Artificial General Intelligence* (B. Goertzel, A. I. Panov, A. Potapov, and R. Yampolskiy, eds.), vol. 12177, pp. 95–106, Springer, 2020.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1135–1144, 2016.

[7] D. A. Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *Advances in Neural Information Processing Systems*, pp. 7775–7784, 2018.

[8] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[9] F. Doshi-Velez and B. Kim, "Considerations for evaluation and generalization in interpretable machine learning," in *Explainable and Interpretable Models in Computer Vision and Machine Learning* (H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven, eds.), pp. 3–17, Springer, 2018.

[10] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 27–30, 2000.

[11] A. Fabregat, S. Jupe, L. Matthews, K. Sidiropoulos, M. Gillespie, P. Garapati, R. Haw, B. Jassal, F. Korninger, B. May, *et al.*, "The reactome pathway knowledgebase," *Nucleic Acids Res.*, vol. 46, no. D1, pp. D649–D655, 2018.

[12] G. O. Consortium, "The gene ontology (go) database and informatics resource," *Nucleic Acids Res.*, vol. 32, no. suppl_1, pp. D258–D261, 2004.

[13] B. Snel, G. Lehmann, P. Bork, and M. A. Huynen, "String: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene," *Nucleic Acids Res.*, vol. 28, no. 18, pp. 3442–3444, 2000.

[14] M. K. Yu, J. Ma, J. Fisher, J. F. Kreisberg, B. J. Raphael, and T. Ideker, "Visible machine learning for biomedicine," *Cell*, vol. 173, no. 7, pp. 1562–1565, 2018.

[15] T. Kang, W. Ding, L. Zhang, D. Ziemek, and K. Zarringhalam, "A biological network-based regularized artificial neural network model for robust phenotype prediction from gene expression data," *BMC Bioinform.*, vol. 18, no. 1, p. 565, 2017.

[16] V. Bourgeais, F. Zehraoui, M. Ben Hamdoune, and B. Hanczar, "Deep GONet: self-explainable deep neural network based on gene ontology for phenotype prediction from gene expression data," *BMC Bioinform.*, vol. 22, no. 10, p. 455, 2021.

[17] H. A. Elmarakeby, J. Hwang, R. Arafeh, J. Crowdis, S. Gang, D. Liu, S. H. AlDubayan, K. Salari, S. Kregel, C. Richter, T. E. Arnoff, J. Park, W. C. Hahn, and E. M. Van Allen, "Biologically informed deep neural network for prostate cancer discovery," *Nature*, vol. 598, no. 7880, pp. 348–352, 2021.

[18] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of International Conference on Machine Learning*, vol. 70, pp. 1263–1272, 2017.

[20] S. Jin, X. Zeng, F. Xia, W. Huang, and X. Liu, "Application of deep learning methods in biological networks," *Brief. Bioinformatics*, vol. 22, no. 2, pp. 1902–1917, 2021.

[21] S. Rhee, S. Seo, and S. Kim, "Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3527–3534, 2018.

[22] R. Ramirez, Y.-C. Chiu, A. Hererra, M. Mostavi, J. Ramirez, Y. Chen, Y. Huang, and Y.-F. Jin, "Classification of cancer types using graph convolutional neural networks," *Front. Phys.*, vol. 8, no. 203, p. 203, 2020.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of International Conference on Learning Representations*, 2017.

[24] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, vol. 29, pp. 3837–3845, 2016.

[25] A. Torrente, M. Lukk, V. Xue, H. Parkinson, J. Rung, and A. Brazma, "Identification of Cancer Related Genes Using a Comprehensive Map of Human Gene Expression," *PLOS ONE*, vol. 11, no. 6, p. e0157484, 2016.

[26] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, "The cancer genome atlas (TCGA): an immeasurable source of knowledge," *Contemp Oncol*, vol. 19, no. 1A, pp. 68–77, 2015.

[27] S. W. Lowe and A. W. Lin, "Apoptosis in cancer," *Carcinogenesis*, vol. 21, no. 3, pp. 485–495, 2000.

[28] D. Dernoncourt, B. Hanczar, and J.-D. Zucker, "Experimental analysis of feature selection stability for high-dimension and low-sample size gene expression classification task," in *2012 IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE)*, pp. 350–355, 2012.