BIBS seminar

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

Taewan Goo

Bioinformatics, 38(9), 2022, 2504–2511
https://doi.org/10.1093/bioinformatics/btac147
Advance Access Publication Date: 10 March 2022
Original Paper

Gene expression

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

Victoria Bourgeais *, Farida Zehraoui and Blaise Hanczar*

Computer Science Department, IBISC, Université Paris-Saclay (Univ. Évry), Évry-Courcouronnes 91020, France

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

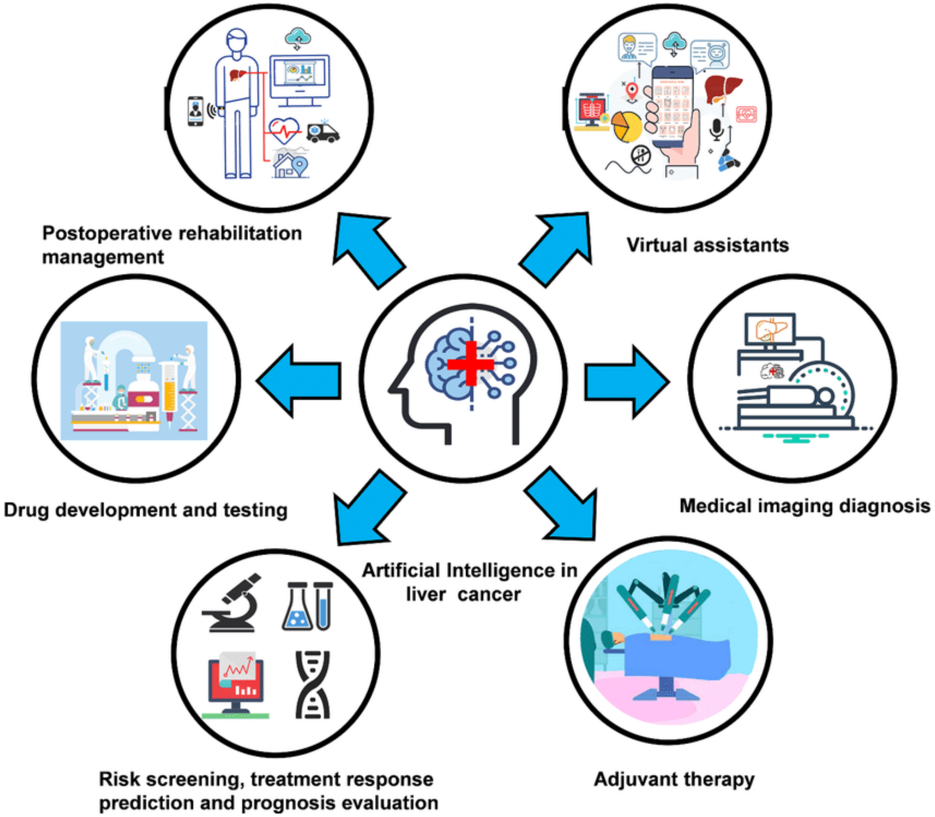
Received on November 21, 2021; revised on February 2, 2022; editorial decision on February 27, 2022; accepted on March 7, 2022

Contents

1. Introduction
2. Graph-GONet
3. Results
4. Summary

Introduction

Precision Medicine


Postoperative rehabilitation management
Virtual assistants
Drug development and testing
Medical imaging diagnosis
Artificial Intelligence in liver cancer
Risk screening, treatment response prediction and prognosis evaluation
Adjuvant therapy

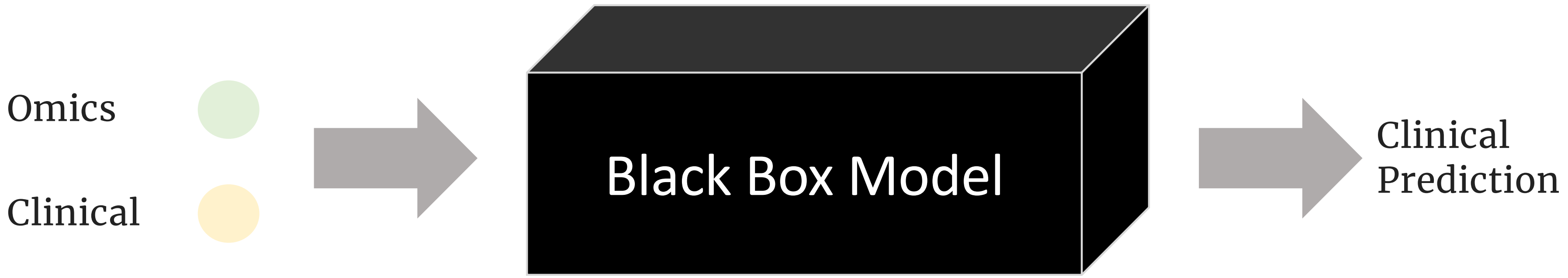
- The objective of precision medicine is to propose medical solutions at different stages of the health care (**diagnosis, prognosis, treatment**).
- These medical solutions take into consideration the unique low-scale characteristics of the patients, known as **omics profile.**

Deep Learning in Biology

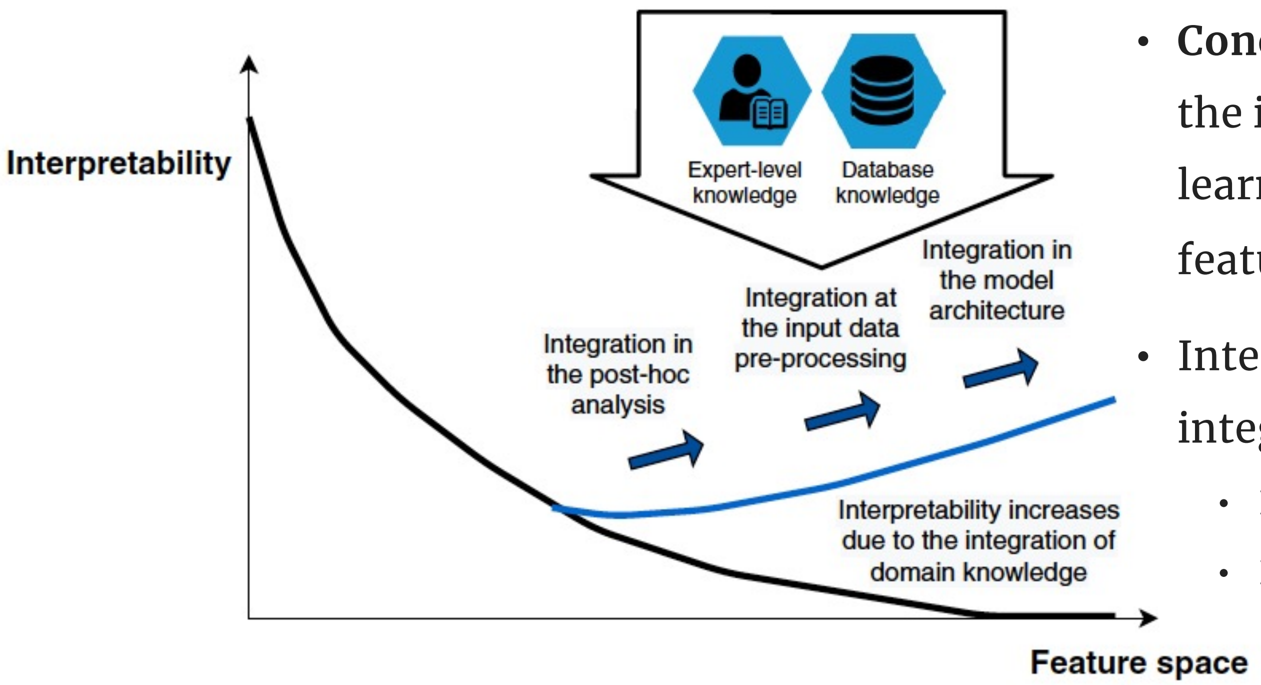
- There is an increasing interest in the use of Deep Learning (DL) based methods as a supporting analytical framework in oncology.
- DL methods in oncology leverage multi-omics data and advanced algorithms, such as GCNs and CNNs, for enhanced cancer subtype classification, survival prediction, and drug response analysis.
- These techniques facilitate precise patient stratification and identification of novel biomarkers, driving the progression towards more tailored and effective cancer treatments in the realm of precision and personalized medicine.

Challenge of Interpretability in ML

- One of the most challenging problems that prevent the development of ML in healthcare is its lack of interpretability.
- In fact, most ML algorithms, including DL approaches, are considered black boxes.
- This means that these models do not provide an explanation to users of their complex decision-making process, only their final prediction.
- Thus, an important issue today is making ML algorithms interpretable.



Challenge of Interpretability in ML


Interpretability
Expert-level knowledge
Database knowledge
Integration in the model architecture
Integration at the input data pre-processing
Integration in the post-hoc analysis
Interpretability increases due to the integration of domain knowledge
Feature space

- **Conceptual framework** for enhancing the interpretability of machine learning models in the context of feature space.
- Interpretability increases due to the integration of domain knowledge
  - Expert level knowledge
  - Database knowledge

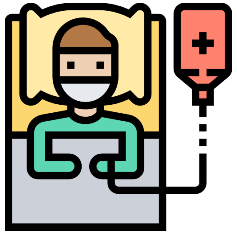
Explainability vs. Interpretability

- **Explainability** is often used interchangeably with interpretability, however the distinction must be made as the **Explainability is product by the Interpretability.**
- **Explainability** refers to a **collection of features from the interpretable domain** that contributes to the production of an abstract statement.
- **Interpretability** refers to mapping this statement into the domain the **human expert can perceive, comprehend, and understand** [1].

[1] Holzinger A, Müller H. Toward Human–AI interfaces to support explainability and causability in medical AI. Com- puter. 2021;54(10):78–86. https://doi.org/10.1109/MC.2021.3092610.

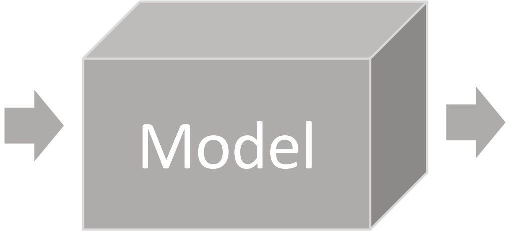
An Example of Explainability vs. Interpretability

Patient A



Gene X (DNA): mutated

Gene Y (RNA) : high expression



Sensitive group for Drug Z

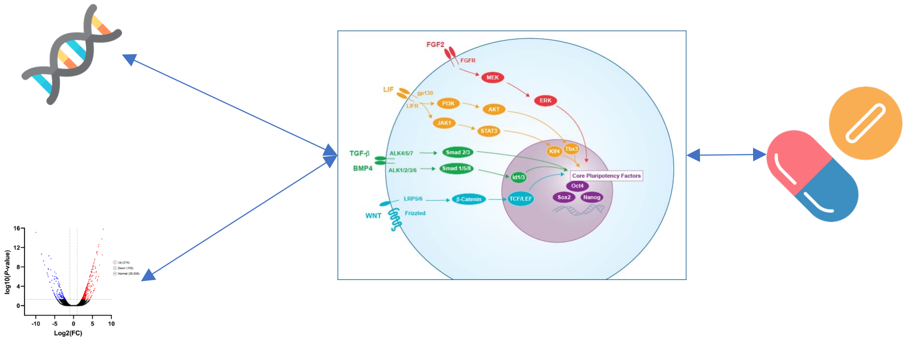
Explainability

"Patients with mutation X and high expression of gene Y tend to have a better response to drug Z."

Interpretability

" This might involve explaining that **mutation X affects a particular signaling pathway** known to be **involved in drug Z's mechanism of action**, and **high expression of gene Y** is a marker of **a subtype of the disease** that is known to be **sensitive to drug Z.**"

Gene X



Gene Y

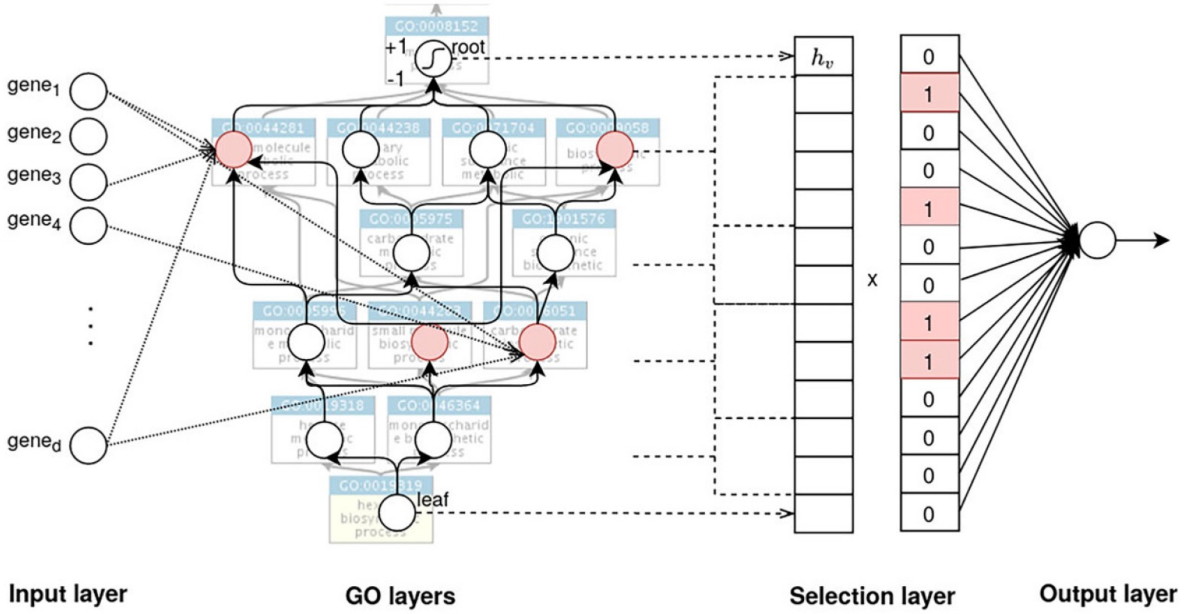
Importance of Interpretability in Medical Field

- Especially in the **medical field**, understanding **why a phenotype** has been predicted is necessary to ensure that a **prediction is based on reliable medical features** rather than on irrelevant artifacts for its end users (e.g. **researchers, clinicians, patients**).
- Regardless of the model's effectiveness, this will affect an **end user's decisions** and **confidence toward the model**.
- Finally, a detailed print-out of a model's **decision-making process** may reveal a **new biological signature** that would otherwise remain undetected for an end user to investigate further.

GraphGONet

Introduction to GraphGONet

- **GraphGONet** is introduced as a **self-explaining neural network** specifically designed to **integrate knowledge from the Gene Ontology (GO)** for phenotype prediction based on **gene expression** profiles of patients.



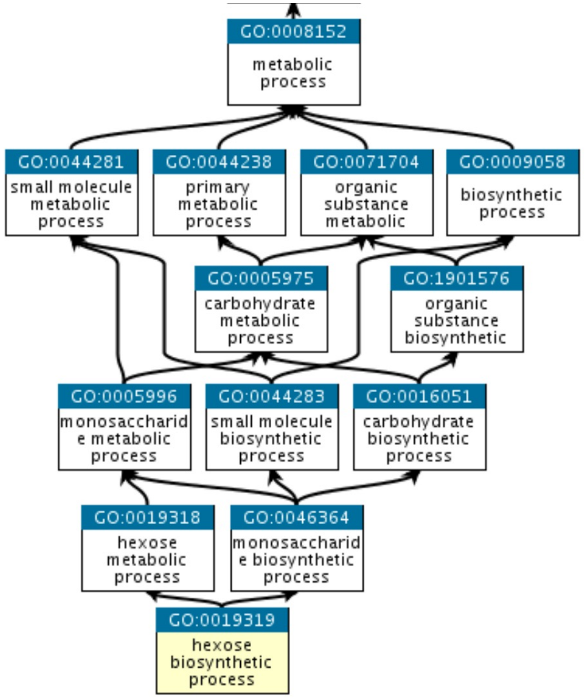
Background: Gene Ontology

- The Gene Ontology (GO) is a major bioinformatics initiative that aims to **standardize the representation of gene and gene product** attributes across species and databases.
- It provides a controlled vocabulary of terms for describing gene product characteristics and gene product annotation data from GO Consortium member databases.

GENEONTOLOGY
Unifying Biology

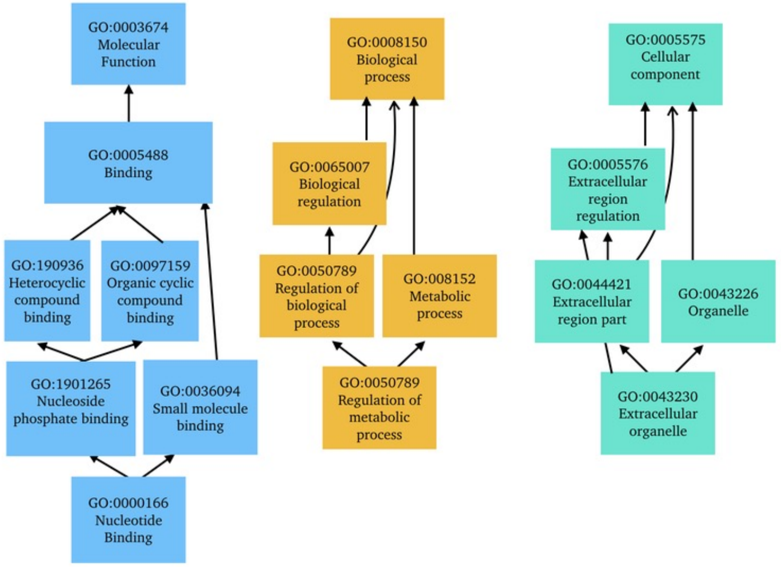
Background: Gene Ontology

- Overview of GO
  - **GO Identifier**: Each box represents a term within the GO, and each term is identified by a unique identifier, such as “GO:0008152”. Each GO Identifier is used to index and reference specific biological processes, molecular functions, or cellular components.
  - **Hierarchical GO Terms**: Each term describe a specific biological concept. GO terms are hierarchically structured; more general terms are found at the top, while more specific terms are found at the lower levels.
  - Categories of GO: There are three main categories of GO: **Biological process**, **Molecular functions**, and **Cellular Components.**


GO:0008152
metabolic process
GO:0044281
small molecule metabolic process
GO:0044238
primary metabolic process
GO:0071704
organic substance metabolic
GO:0009058
biosynthetic process
GO:0005975
carbohydrate metabolic process
GO:1901576
organic substance biosynthetic
GO:0005996
monosaccharide metabolic process
GO:0044283
small molecule biosynthetic process
GO:0016051
carbohydrate biosynthetic process
GO:0019318
hexose metabolic process
GO:0046364
monosaccharide biosynthetic process
GO:0019319
hexose biosynthetic process

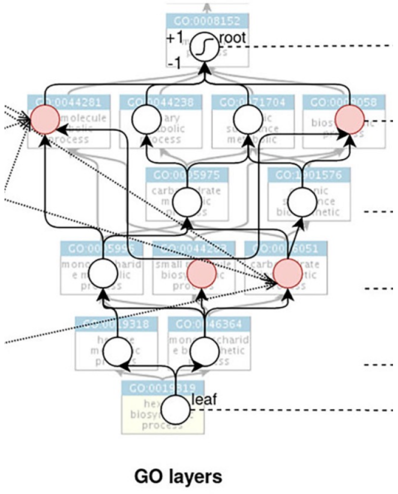
Background: Gene Ontology

- Three categories of GO
  - **Biological Process**: this domain encompasses processes and series of events performed by one or more gene products, such as a cellular process or signaling pathway.
  - **Molecular Function**: this domain describes the elemental activities of a gene product at the molecular level, such as binding or catalysis.
  - **Cellular Component**: this domain covers where gene products are active, suggesting the parts of a cell or its extracellular environment.


GO:0003674 Molecular Function
GO:0005488 Binding
GO:190936 Heterocyclic compound binding
GO:0097159 Organic cyclic compound binding
GO:1901265 Nucleoside phosphate binding
GO:0036094 Small molecule binding
GO:0000166 Nucleotide Binding
GO:0008150 Biological process
GO:0065007 Biological regulation
GO:0050789 Regulation of biological process
GO:008152 Metabolic process
GO:0050789 Regulation of metabolic process
GO:0005575 Cellular component
GO:0005576 Extracellular region regulation
GO:0044421 Extracellular region part
GO:0043226 Organelle
GO:0043230 Extracellular organelle

Self-Explainability of GraphGONet

- GraphGONet achieves **self-explainability** in its deep learning model through the following features:
  - **Layered Architecture Mimicking GO:** The hidden layers of GraphGONet are organized to mimic the architecture of the **Gene Ontology**.



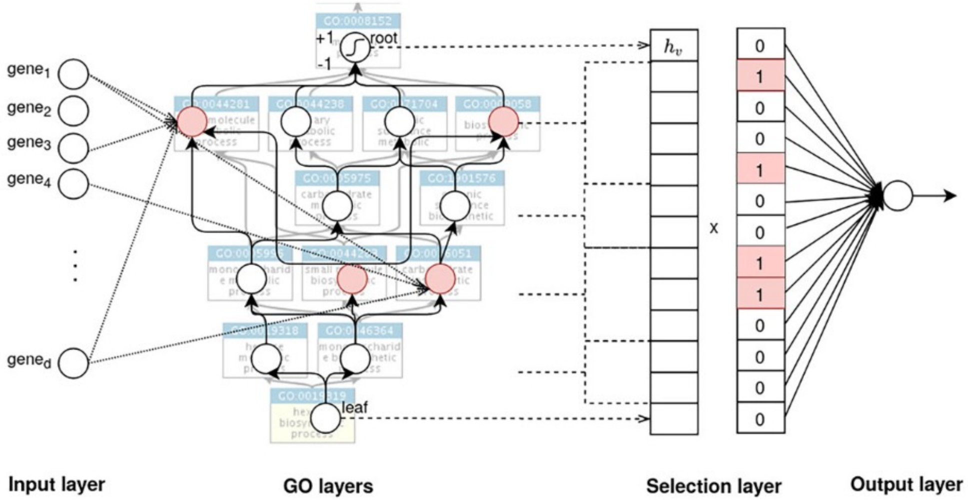
  - **Neuron representing GO terms:** Each neuron in these hidden layers represents a specific GO term. The connections between these neuron represent the relationships between different GO terms. These connections are oriented from lower to upper GO levels in the hierarchy.
  - **Selective Connectivity Based on GO Relationships:** A neuron corresponding to a specific GO term is connected only to those genes and child neuron that are **directly associated with that term. This selective connectivity** ensures that the propagation of information through the **network accurately reflects the biological relationships** encoded in the Gene Ontology

Self-Explainability of GraphGONet

- **Propagation of Gene Expression Information**
  - The input layer receives the gene expression profile of a patient.
  - This information is then propagated through the network. For each neuron (representing a GO term), its activation value is computed based on the GE profile restricted to the genes associated with that GO term and the activations of its child nodes.
  - This process ensures that the activation of each node reflects both the direct gene expression data relevant to its GO term and the information from related terms.
- **Capturing Biological Relationships**
  - Through this structure and process, GraphGONet captures the **complex biological relationships encoded in the Gene Ontology.**
  - This allows the network to **integrate and interpret the GE data** in the context of **established biological knowledge.**

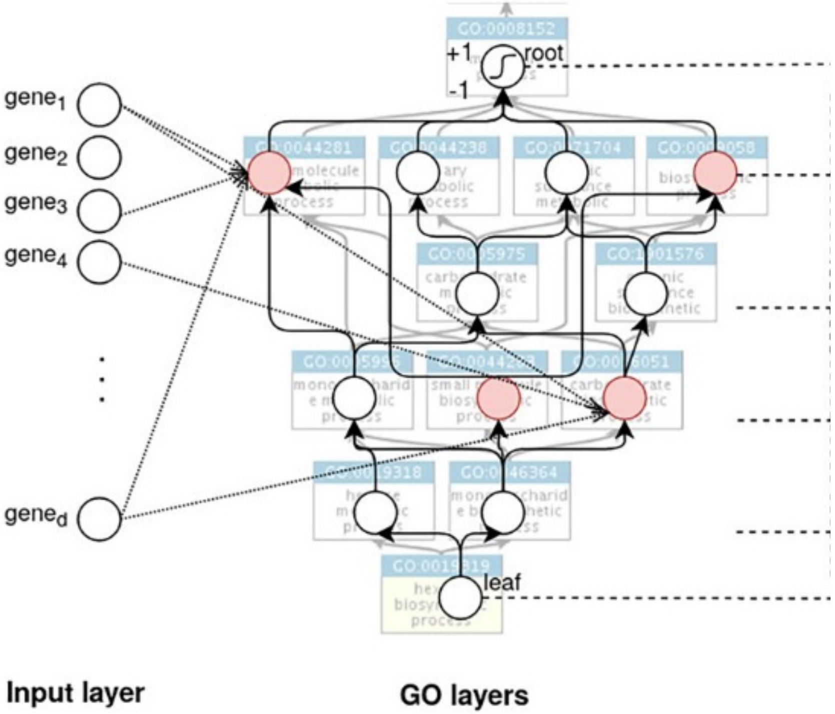
Architecture of GraphGONet

- Let (X, Y) be a training example, where $X = [X_1, \dots, X_d]$ is the GE 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
  - $y_c = 0$ otherwise



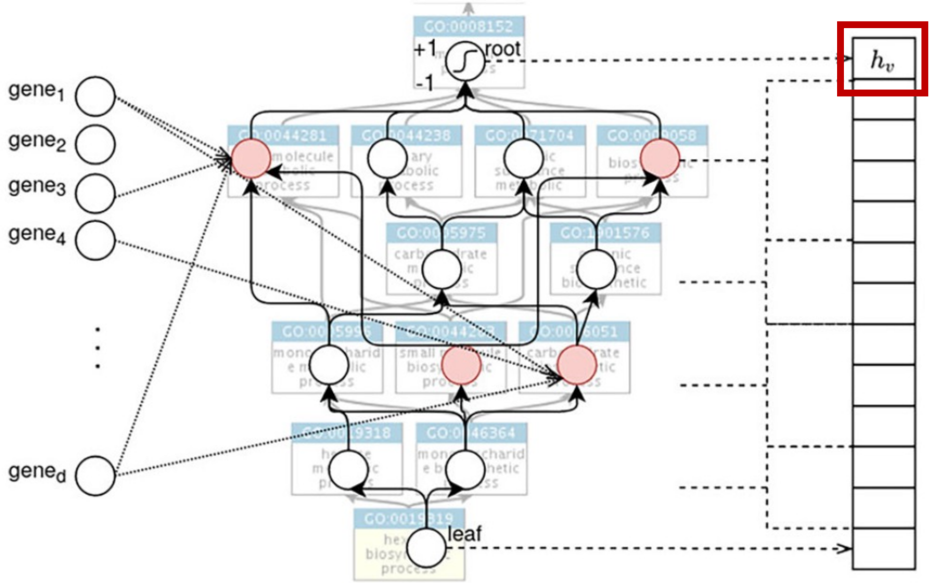
Architecture of GraphGONet

- 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, and the last hidden layer represent the root.



Architecture of GraphGONet

- Let $\boldsymbol{G}(\boldsymbol{v})$ be the **set of genes** associated with a GO term corresponding to a **neuron $\boldsymbol{v}$** in GraphGONet **and $\boldsymbol{N}(\boldsymbol{v})$** the set of neurons corresponding to the **children of the neuron $\boldsymbol{v}$.**
- The activation value of 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 $N(v)$.



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

where $w_G, w_N \in \mathbb{R}$ are trainable parameters shared by all nodes v, σ is the tanh activation function.

Architecture of GraphGONet

- The activation value of neuron h_v :

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

Shared parameters

- Embedding of the expression of gene set:

$$h_{G(v)} = W_v X_{G(v)} + b_v, \text{where}(\ W_v \in \mathbb{R}^{|G(v)|}, b_v \in \mathbb{R})$$

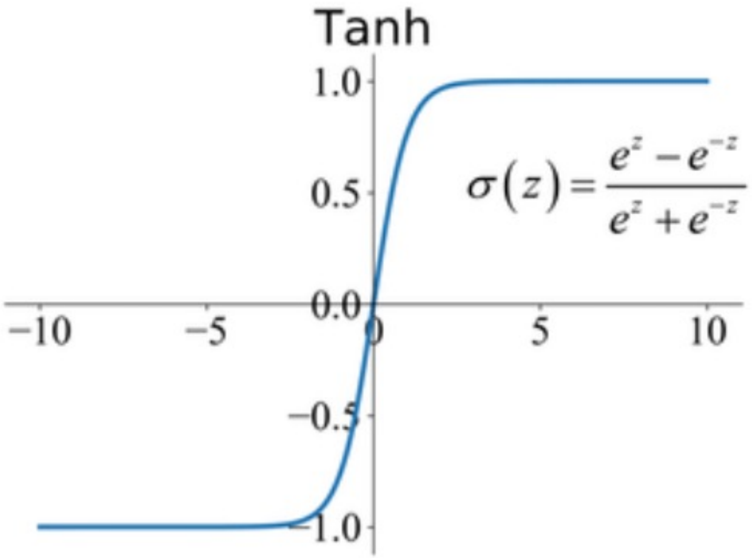
Neuron specific parameters

- Embedding of the activation of the neuron set $N(v)$:

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

Architecture of GraphGONet

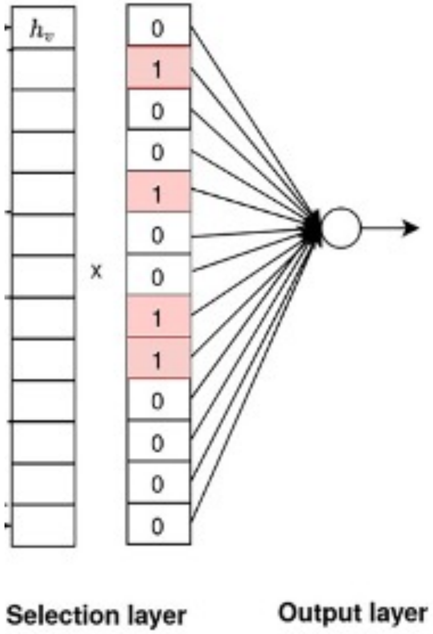
- Activation function – tanh
  - In this paper, the choice of the tanh function is more relevant than the ReLU one.
  - The tanh function will saturate the neurons selected in the next part of the network (the selection layer), to values close to +1 to −1.
  - It makes the interpretation of the prediction much easier.



Link: https://paperswithcode.com/method/tanh-activation

Architecture of GraphGONet

- The next part of the model is the selection of the most activated neurons in absolute value.
  - Their associated GO terms will be used to support the explanation of a prediction.
  - The process consist of:
    - Concatenating the activation of all neurons, except those of the input layer.
    - Computing a mask M identifying the most activated neurons $M_v = 1$ if $v \in top(r)$, $M_v = 0$ otherwise, where r is the selection ratio and top is a function returning the indicies of the K_r neurons selected.
    - 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.



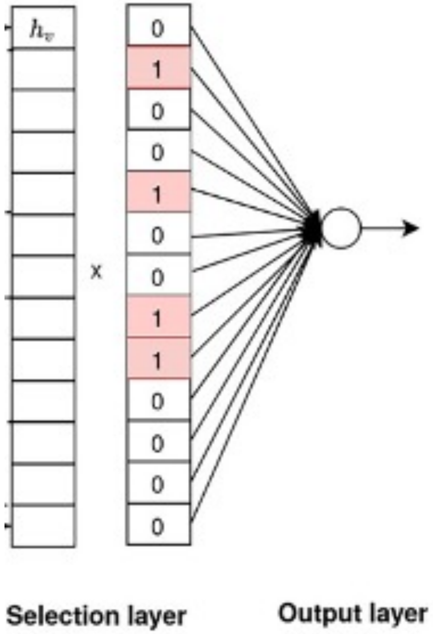
Architecture of GraphGONet

- The last layer returns the output, where each neuron represents one of the C classes.
- It is a linear combination of the output is computed from:

$$z_c = \sum_{j=1}^{K} h_{select,j}\, w_{jc} + b_c, \text{ where } (W \in \mathbb{R}^{K\times C}, b \in \mathbb{R}^{C})$$

- The output activations are transformed into probabilities using the softmax function:

$$O_c = \frac{\exp(z_c)}{\sum_{j=1}^{C} \exp(z_i)}$$



Model Explainability

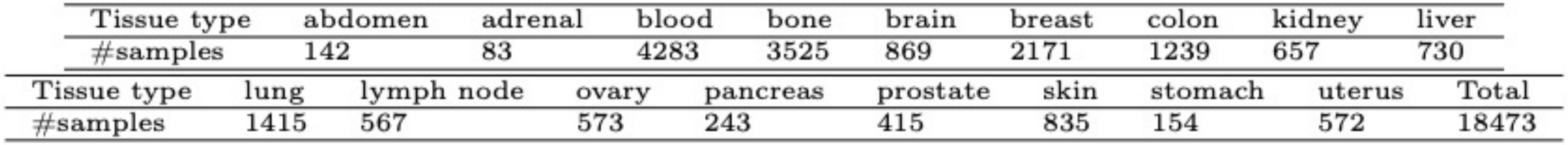
- The model automatically provides both a prediction and an explanation for a given patient.
  - The explanation takes the form of a list of GO terms implied in the final computation of the prediction, with their score of importance.
  - The number of GO terms in the list is determined by the selection ratio r.
  - 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:

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

Results

Datasets

- Gene Expression Datasets
  - ArrayExpress Database (E-MTAB-3732)
    - **Data**: Heterogeneous microarray data from around 40,000 Affymetrix HG-U133Plus2 chip arrays.
    - **Composition**: After quality control and normalization, includes 54,675 input probes for 27,887 cancer and non-cancer samples from 17 different tissue types.
    - **Data split**: 80% for training, 20% for testing, maintaining original proportions (66% cancer, 34% non-cancer).

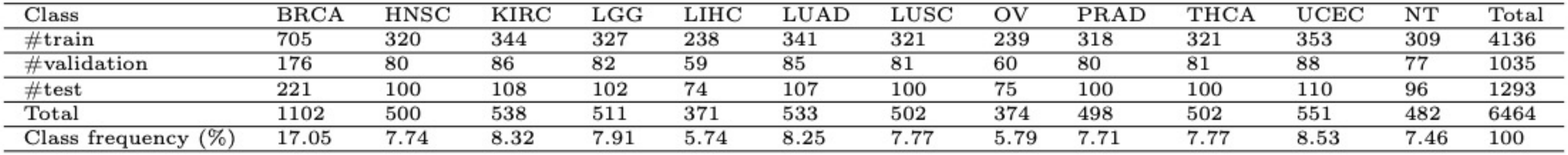


Tissue type	abdomen	adrenal	blood	bone	brain	breast	colon	kidney	liver
#samples	142	83	4283	3525	869	2171	1239	657	730

Tissue type	lung	lymph node	ovary	pancreas	prostate	skin	stomach	uterus	Total
#samples	1415	567	573	243	415	835	154	572	18473

Datasets

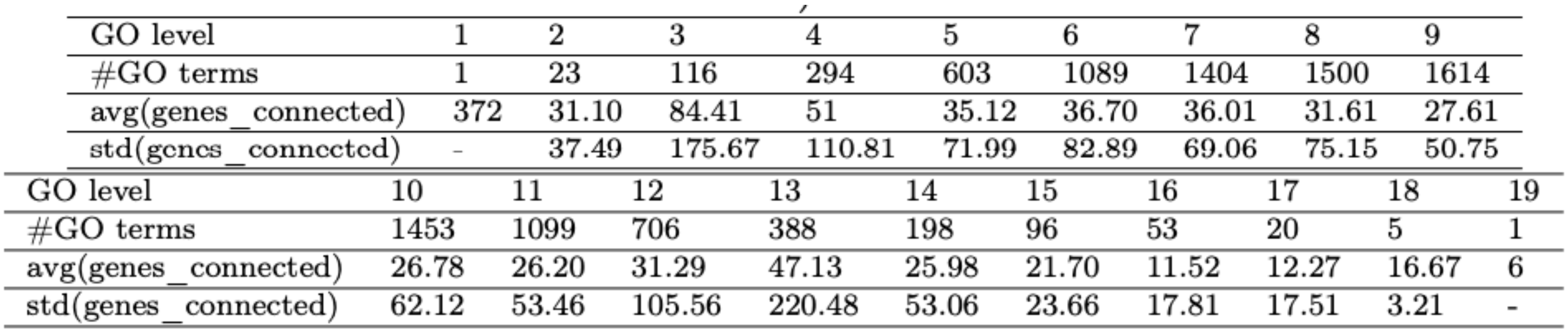
- Gene Expression Datasets
  - The Cancer Genome Atlas (TCGA) – RNA-seq
    - **Data**: Includes 5,892 cancer samples across 11 cancer types and 482 non-cancer samples, totaling 56,602 input genes.
    - **Data split**: 80% for training, 20% for testing, maintaining original proportions.



Class	BRCA	HNSC	KIRC	LGG	LIHC	LUAD	LUSC	OV	PRAD	THCA	UCEC	NT	Total
#train	705	320	344	327	238	341	321	239	318	321	353	309	4136
#validation	176	80	86	82	59	85	81	60	80	81	88	77	1035
#test	221	100	108	102	74	107	100	75	100	100	110	96	1293
Total	1102	500	538	511	371	533	502	374	498	502	551	482	6464
Class frequency (%)	17.05	7.74	8.32	7.91	5.74	8.25	7.77	5.79	7.71	7.77	8.53	7.46	100

Choice of the GO layers

- In these experiments, only the **biological process** subontology (GO-BP) of the GO is integrated into GraphGONet.
  - GO-BP was chosen as it is often preferred by biologists for explaining predictions.
  - The GO version used dates from 01-06-2020 and contains originally 29,112 GO-BP terms.
  - The directed acyclic graph is organized into levels, where the level of a GO term is determined according to its longest path with the root.



GO level	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
#GO terms	1	23	116	294	603	1089	1404	1500	1614	1453	1099	706	388	198	96	53	20	5	1
avg(genes_connected)	372	31.10	84.41	51	35.12	36.70	36.01	31.61	27.61	26.78	26.20	31.29	47.13	25.98	21.70	11.52	12.27	16.67	6
std(genes_connected)	-	37.49	175.67	110.81	71.99	82.89	69.06	75.15	50.75	62.12	53.46	105.56	220.48	53.06	23.66	17.81	17.51	3.21	-

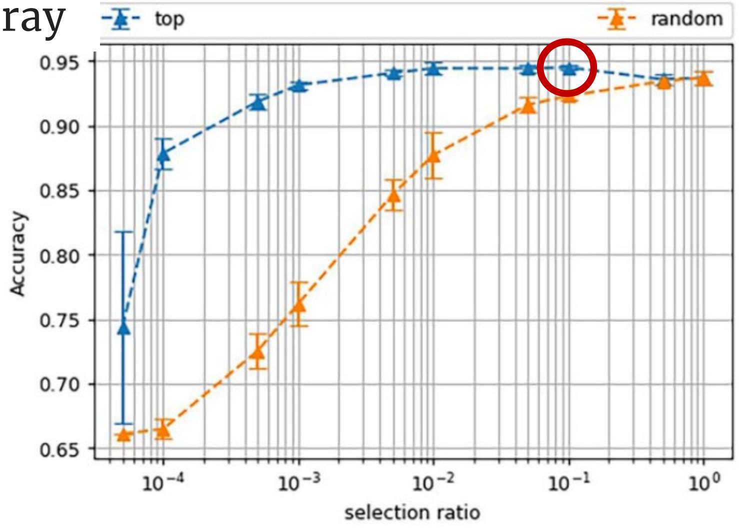
Sensitivity Analysis

- In the first experiment, they analyze 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 outcome.
- They evaluate the selection process and the value of the hyperparameter r and compare this process with a random selection.
    - They 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 biases.

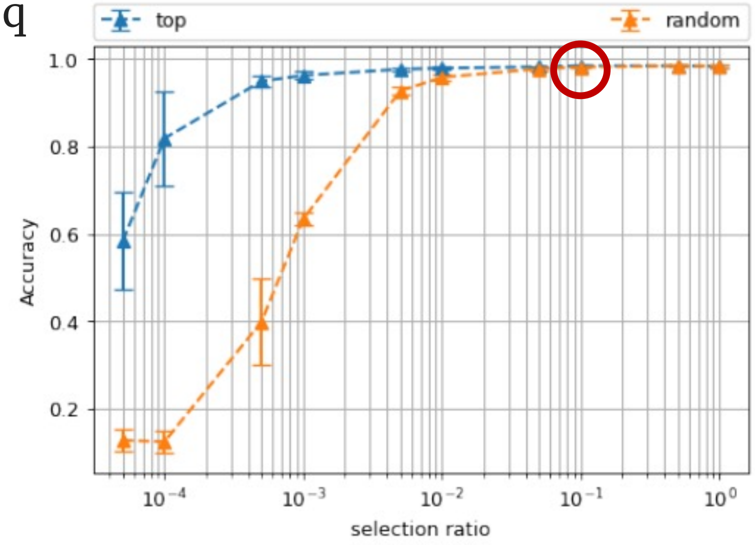
Sensitivity Analysis

- The average and the standard deviation of the models ’ accuracy are reported according to the value of r in right figures.
- They finds that ‘top’ selection generally outperforms random selection in both datasets used for cancer diagnosis.
- The best performance is achieved with ‘top’ selection at a ratio of 0.1 in both datasets.
- Interestingly, optimal predictions are based on a small proportion of neurons (around 1,000).

Microarray



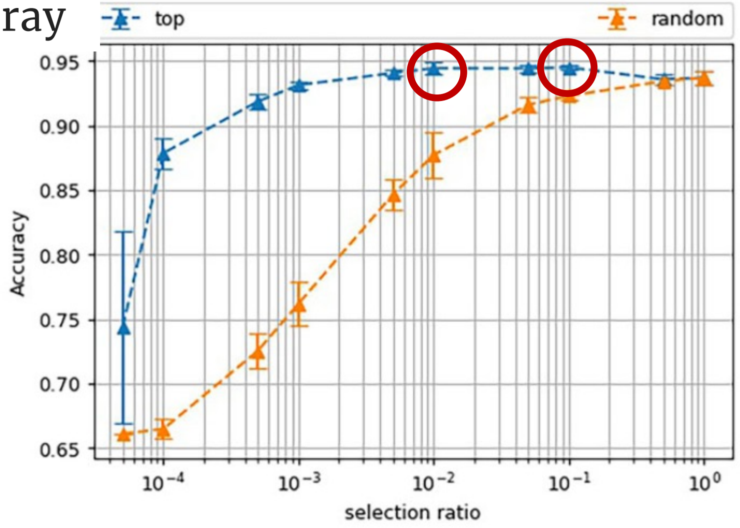
RNA-seq



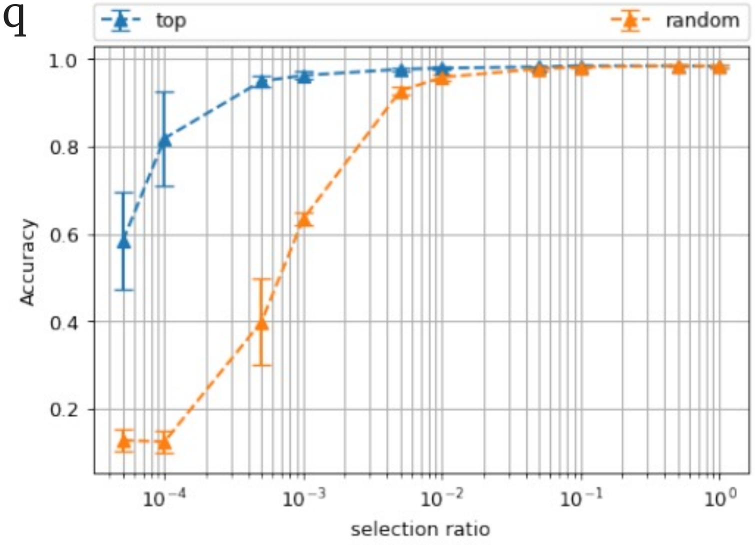
Sensitivity Analysis

- The first model, with the best performance, used a **ratio (r) of 0.01,** selecting **around 100** Gene Ontology (GO) terms.
    - Despite not being the ratio with the best average performance, the difference between r = 0.01 and r = 0.1 is negligible (<0.005) for both datasets.
- The second model chose r = 0.001 for a reasonable balance between performance and interpretability, with a slight accuracy decrease of about 1.5% while reducing the number of selected GO terms to **around 10.**

Microarray



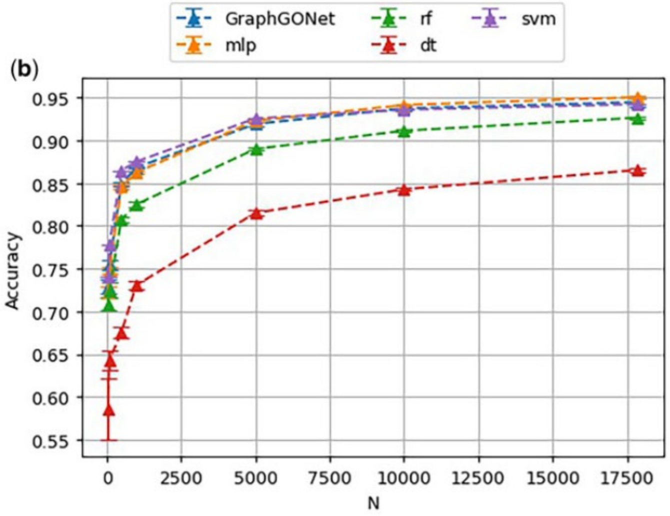
RNA-seq



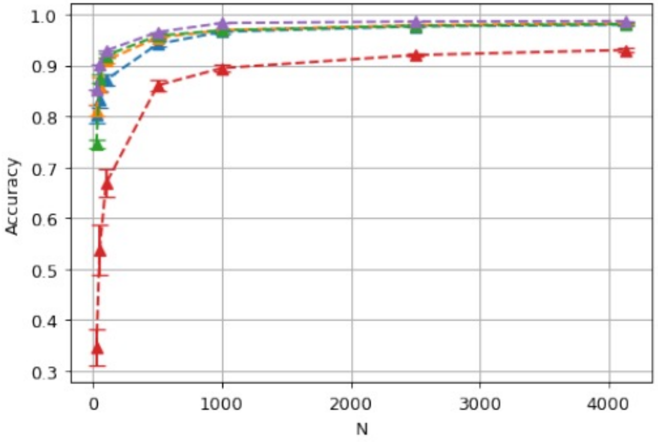
Sensitivity Analysis

- Comparison with classical machine learning algorithms:
  - In another experiment, a **proposed model (at r = 0.01)** was compared with standard machine learning algorithms.
  - These methods were trained on varying sizes of training sets, from full size to minimal (50 samples for the microarray dataset and 25 for the TCGA dataset).
  - The study observed that the best accuracies were achieved with the highest number of samples, with deep learning methods and support vector machines performing similarly in both datasets.
  - GraphGONet proved to be as **competitive as non-explainable ML** and **deep learning algorithms** and clearly outperformed the only comparable **explainable method (decision tree)**, regardless of the training set size.

Microarray

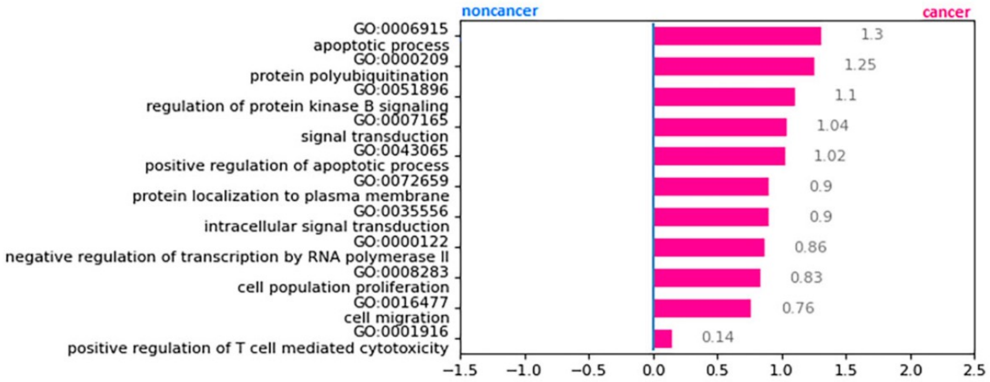


RNA-seq



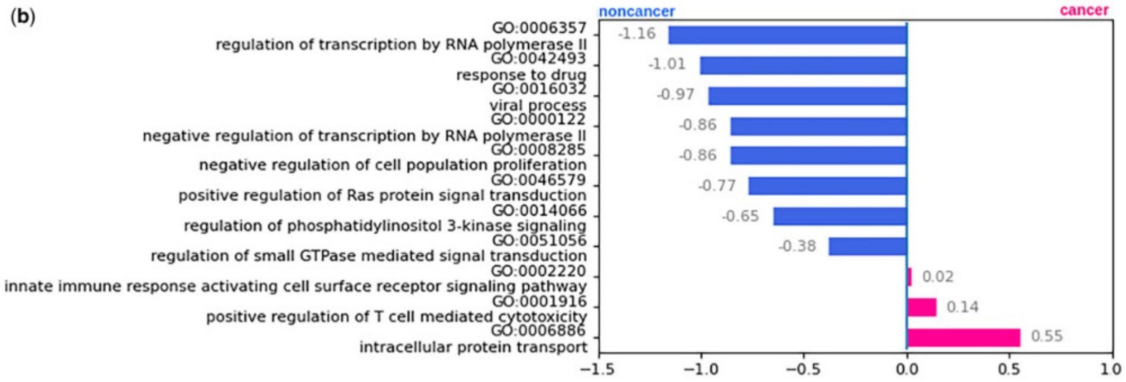
Interpretation of a patient outcome

- They used model (at ratio of 0.001), resulting in it utilizing **only 11 neurons** and their associated **GO terms.**
  - Each patient's prediction is based on different subsets of these 11 GO terms.
- The selected GO terms typically belong to **intermediate levels** (between levels 6 and 10).
- The relevance score of each GO term is computed to identify the most influential terms in the subset.
- This score indicates how significantly a GO term impacts the final prediction.



Sample correctly predicted cancer with a probability of 1 and a total relevance score of 10.58

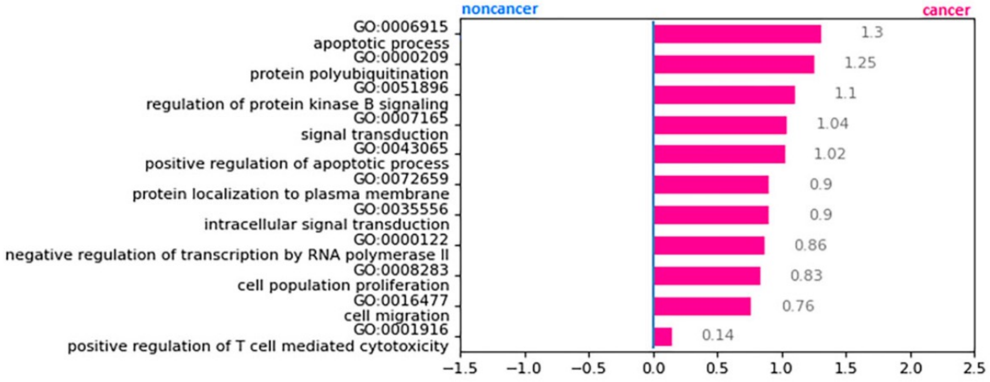
(b)



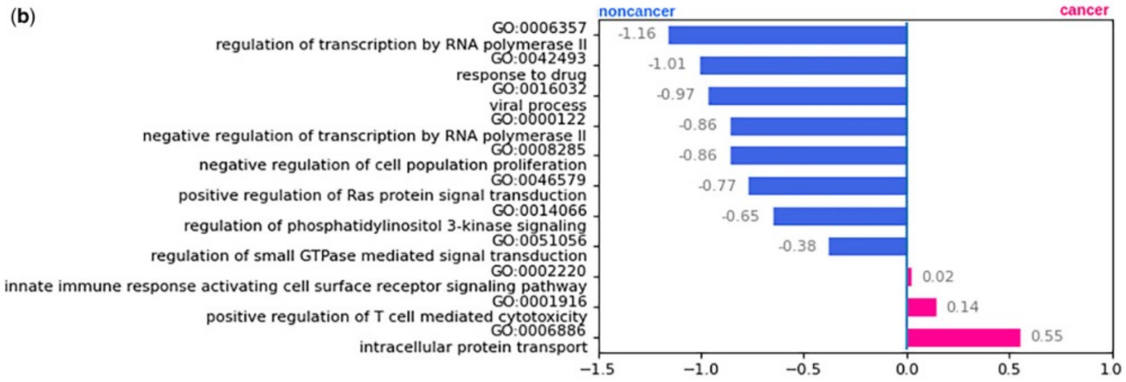
Sample correctly predicted noncancer with a probability of 0.996 and a total relevance of -5.47

Interpretation of a patient outcome

- The 11 GO terms are ranked by their relevance score in descending (ascending) order for cancer patients.
- For instance, in the case of a cancer patient, all GO terms have a positive sign, with ten out of eleven terms having a relevance score close to the average of 0.92.
  - Specific GO terms like **GO: 0006915** and **GO: 0043065**, related to apoptosis, are identified as playing a role in cancer.
- The relevance score helps in discerning the effective impact of GO terms on the final prediction and **quantifying prediction uncertainty**.



Sample correctly predicted cancer with a probability of 1 and a total relevance score of 10.58



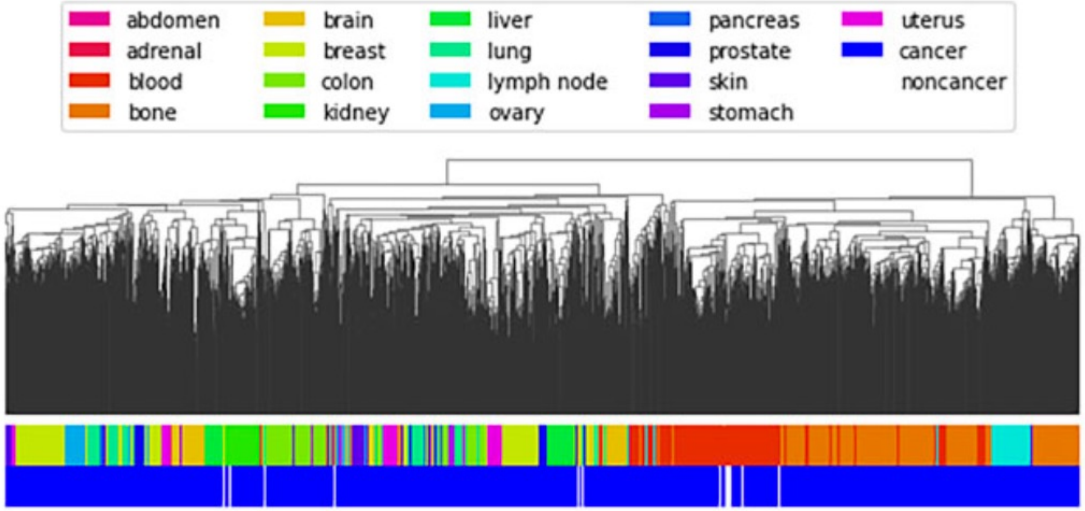
Sample correctly predicted noncancer with a probability of 0.996 and a total relevance of -5.47

Interpretation of the model

- The approach involves measuring the **similarity of explanations** between patients by analyzing clustering based on relevance profiles.
- Relevance matrices of size (N, K), where **N is the number of samples and K is the number of GO terms**, are created for test samples.
- Each row in these matrices represents a patient's relevance profile.

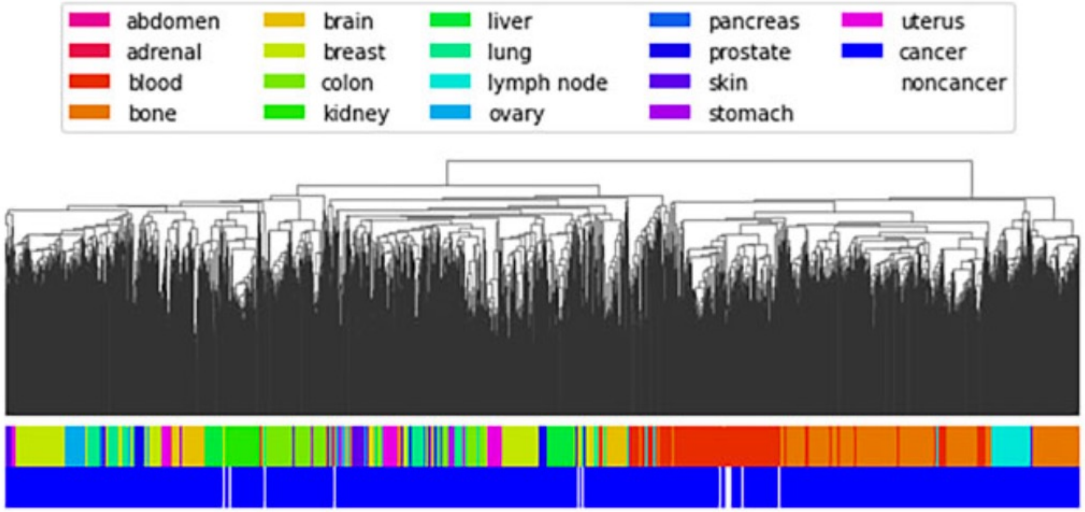
Interpretation of the model

- Hierarchical clustering is applied to these matrices, using **average linkage criteria** and **Euclidean distance as the metric**.
- Dendrograms are used to illustrate this clustering.
- The dendrogram reveals clusters grouping patients from the same tissues, suggesting that certain neurons and corresponding GO terms can extract cancer features specific to certain tissue types.



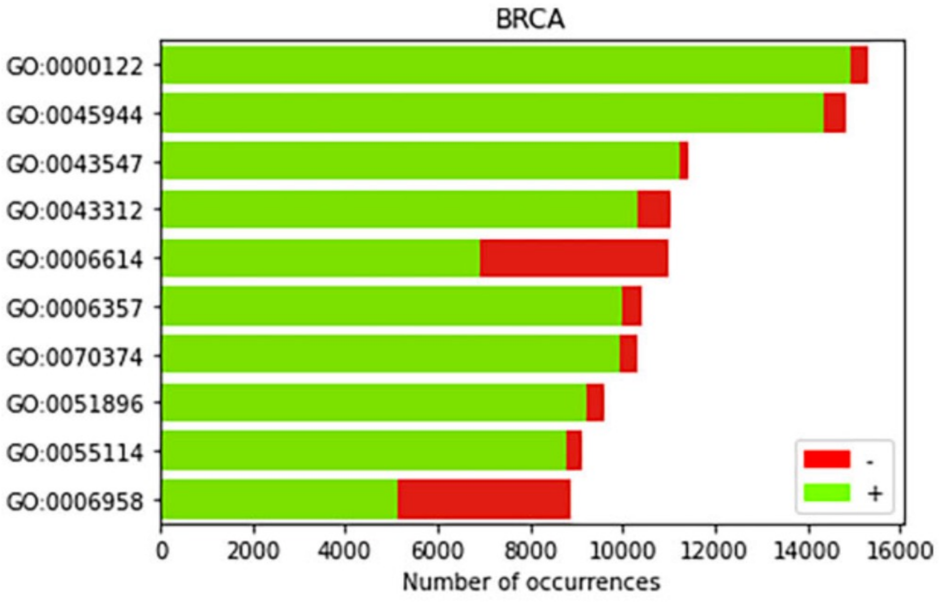
Interpretation of the model

- While the model is not initially designed to predict cancer tissue type, it successfully identifies multiple cancer signatures associated with different tissues.
- Errors in prediction are distributed across clusters, with a notable proportion in the cluster related to blood tissue.



Interpretation of the model

- To evaluate the consistency, **100 GraphGONet models** with a selection ratio of 0.01 are trained.
- **Relevance and occurrence matrices** are computed, indicating whether a GO term has been selected by the selection layer.
- The frequency of GO term selection and the relevance score are analyzed to understand the model's predictions.
- GO terms that are frequently selected are considered to contain relevant biological information.



Summary

GraphGONet's Functionality and Validation

- **GraphGONet** leverages a complete knowledge graph and its semantics to accurately perform prediction tasks.
- It has been **validated on two datasets** and can handle any knowledge represented by a Directed Acyclic Graph (DAG).
- The model introduces **novel elements in sequential propagation** and the **selection layer**, allowing the integration of genomic expression (GE) in end-to-end learning and providing biological insights into decision-making.
- Propagation in Gene Ontology (GO) layers originates from the process used in graph convolutional layers, enabling the **inclusion of all GO levels** and **various connection types between neurons.**

Advantages over traditional neural network

- The number of parameters associated with connections inside hidden GO layers is reduced to **two shared parameters**, in contrast to about **23,900 parameters** for an MLP with skip connections.
- Traditional neural networks often use complex **post hoc methods to estimate gene relevance**, leading to a large set of relevance values and less understandable explanations.
- GraphGONet, on the other hand, offers **accessible and understandable explanations** to biological experts by producing a small set of GO terms with associated relevance scores.
- GraphGONet's **biological explanations are stable**, as **evidenced by similar sets of GO terms returned across 100 models** with different parameter initializations.
- The model's explanations are based on **higher semantic concepts (GO terms instead of genes)**, adding stability to the interpretations provided.
- The relevance score of each GO term is easy to compute and effectively quantifies their final contribution in the subset.

Further developments

- Future work includes incorporating other ontologies, **such as pathways**, to enrich biological explanations and satisfy **multimodal causability**.
- There are plans to further **study the model's uncertainties** and develop a more rigorous **quantitative estimation** of GraphGONet's **system reliability.**

Thank you