Interprétation de l'apprentissage profond pour la prédiction de phénotypes à partir de données d'expression de gènes

Victoria Bourgeais

université PARIS-SACLAY

Interprétation de l'apprentissage profond pour la prédiction de phénotypes à partir de données d'expression de gènes

Interpretation of deep learning for phenotype prediction from gene expression data

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique,
Référent : Université d'Évry Val d'Essonne

Thèse préparée dans l'unité de recherche IBISC (Université Paris-Saclay, Univ Evry), sous la direction de Blaise HANCZAR, Professeur, le co-encadrement de Farida ZEHRAOUI, Maîtresse de conférence

Thèse soutenue à Paris-Saclay, le 14 octobre 2022, par

Victoria BOURGEAIS

Composition du jury

Antoine Cornuéjols Professeur, AgroParisTech, LINK	Président
Grégoire Montavon Chercheur associé, TU Berlin, IDA	Rapporteur & Examinateur
Jean-Philippe Vert Chercheur associé, Mines ParisTech, CBIO	Rapporteur & Examinateur
Florence D'Alché Buc Professeure, Télécom Paris, LTCI	Examinatrice
Flora Jay Chargée de recherche, CNRS, LISN	Examinatrice
Blaise Hanczar Professeur, Université Paris-Saclay, IBISC	Directeur de thèse

THESE DE DOCTORAT
NNT : 2022UPASG069

Remerciements

Je tiens tout d'abord à remercier mes encadrants de thèse, Blaise Hanczar et Farida Zehraoui, pour m'avoir fait confiance tout le long de ce doctorat et pour leur temps précieux qu'ils m'ont accordé. Malgré la pandémie du COVID, mon doctorat s'est passé dans de très bonnes conditions grâce à leur soutien. Ils ont su me transmettre leur passion de la recherche et me faire évoluer dans ce milieu fort riche. Ils m'ont aussi accordé leur confiance au travers de l'encadrement de travaux dirigés, mais également de stages qui m'ont permis d'aborder des aspects différents de la recherche. Milles mercis à eux.

J'aimerais ensuite remercier l'ensemble des membres de mon jury de thèse, Grégoire Montavon, Jean-Philippe Vert, Antoine Cornuéjols, Florence D'Alché Buc, Flora Jay, pour avoir accepté de prendre leur temps pour relire et juger ma thèse. Je les remercie aussi des discussions très enrichissantes que nous avons eues lors de la soutenance. Merci aussi à l'entreprise Oncodesign pour la collaboration que nous avons pu avoir ensemble. J'en profite également pour remercier mes professeurs de l'UTC, pour la formation solide qu'ils m'ont délivrée, mes collègues du CEA dont Gaël de Chalendar, Olivier Ferret et Youssef Tamaazousti, pour m'avoir donné cet avant-goût de la recherche, Marielle Suchet, Christophe Montagne et Anne-Laure Ligozat, pour m'avoir ouvert les yeux sur la possibilité de concilier le développement soutenable et la recherche en informatique.

Puis, je tiens à remercier l'université d'Evry et l'ensemble du laboratoire IBISC pour m'avoir accueillie depuis mon stage en février 2019 et pour l'expérience que j'y ai pu acquérir, et tout particulièrement les personnes suivantes : la direction du laboratoire et l'ensemble des membres de l'équipe AROBAS, ainsi que Serena Cerrito, Sergiu Ivanov, et Amélie Regnault pour leur partage d'expériences et leurs conseils, Murielle Bourgeois, pour son accueil si chaleureux et son souci de prendre soin de "ses enfants", Ludovic Ishiomin, pour son support technique dans la maintenance des serveurs de calcul indispensables à cette thèse, Tina Issa et Ying Li, mes camarades doctorantes depuis le début, merci pour votre amitié, les doctorants et stagiaires, Louis Becquey, Johan Arcile, Jérémie Pardo, Clément Bertrand, Sophie Paillocher, Constance Creux, Alexandre Heuillet, Manel Koumas, Liping Gao, Jing Li, Peng Hu, Yaxing Pang, Junkang He, Xin Feng, Elies Gherbi, Tien Tai Doan, Mohamed Ben Hamdoune, Léa Boulos, pour les moments conviviaux qu'on a pu passer ensemble sur le temps du midi ou après le travail, ceux qui ont partagé ou partagent actuellement mon bureau, les doctorants, Ahmed Meinesidi, Alice Lacan, Aurélien Beaude, et les stagiaires que j'ai encadrés, Thomas Laurent, Flora Carré, Mewe-Hezoudah Kahanam, pour toutes les discussions enrichissantes qu'on a pu avoir. De nouvelles et profondes amitiés y sont nées et un amour aussi. Je tiens ainsi à remercier mon chéri et mon mari depuis peu, Junkai He. À ses côtés, je me suis sentie plus battante. Il a su m'accompagner, me soutenir jusqu'au bout, m'attendre souvent, et me préparer de si bons petits plats !

Enfin, j'ai une pensée pour toute ma famille proche, et en particulier, mes grands-parents paternels, mes parents et mes frères et sœurs, pour leur soutien dans mes projets, leur réconfort, leur patience et la force qu'ils m'ont transmise. Je remercie infiniment ma grand-mère maternelle, qui m'a transmis son goût des sciences. Mes sentiments vont aussi envers mes amis de longue date, spécifiquement, celles qui ont toujours été là, Chloé, Nina, Élodie, Enora, Jacqueline, et finalement, Clément et Gwen, pour nos désirs communs d'engagement pour un meilleur avenir durable.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations et sigles

ACC Accuracy.
ADN Acide désoxyribonucléique.
AE Auto-encodeur.
ARN Acide ribonucléique.
ARNm Acide ribonucléique messager.
AUC Area under the curve.
AUPRC Area under the precision-recall curve.

CNN Convolutional neural network.

DAE Denoising auto-encoder.
DAG Directed acyclic graph.
DL Deep learning.

FFNN Feedforward neural network.

GAN Generative adversarial network.
GCN Graph convolutional network.
GE Gene expression.
GNN Graph neural network.
GO Gene Ontology.
GO-CAM Gene Ontology Causal Activity Model.

IA Intelligence articielle.
IPP Interactions protéines-protéines.

LRP Layerwise Relevance Propagation.

MCC Coefficient de corrélation de Matthews.
ML Machine learning.
MLP Multilayer perceptron.

NGS Séquençage nouvelle génération.
NN Neural network.

RC Residual connection.
RF Random forest.

SL Self-loop.

SVM Support vector machine.

TAL Traitement automatique du langage.

TCGA The Cancer Genome Atlas.

VAE Variational auto-encoder.

WM Weight mean.

XAI Explainable artificial intelligence.

Chap.

1 Introduction

Contenu du chapitre

1.1 Contexte : l'IA au service de la santé

De nos jours, grâce aux progrès rapides des technologies d'acquisition de données, la collecte massive de données structurées comme non structurées sur des patients est rendue possible. Ces données se répartissent dans les catégories, non exhaustives, suivantes :

- clinique (âge, poids, mesures artérielles...) ;
- imagerie médicale et cellulaire (IRM, scanographies...) ;
- signal (électrocardiogrammes, électroencéphalogrammes) ;
- IoT (capteurs, montres connectées) ;
- profil moléculaire (caractéristiques génétiques et moléculaires).

Face à l'accumulation de ces données (phénomène *big data*), la science des données cherche à en extraire de l'information et des connaissances par des algorithmes spécifiques d'apprentissage automatique (*machine learning* (ML) en anglais). Ces algorithmes sont souvent désignés par le terme d'Intelligence Artificielle (IA) dans le monde médiatique. L'un des enjeux actuels est donc d'appliquer les outils d'IA au domaine de la santé [NVR20]. Le rapport de Cédric Villani de 2018 *Donner un sens à l'intelligence artificielle (IA)* [Vil+18 ; NVR20] a notamment identifié la santé comme un des secteurs d'avenir de l'IA « dans lequel la France doit concentrer ses efforts de développement » [1]. Plusieurs champs d'application sont possibles aussi bien dans l'accompagnement des gestes chirurgicaux via la robotique que dans l'aide à la décision ou l'automatisation de certaines tâches. Certaines applications sont déjà au stade de production, par exemple dans le domaine de l'imagerie médicale [2].

Face à cette diversité des données, dans cette thèse, nous nous sommes focalisés sur des données composant le profil dit moléculaire d'un patient. L'exploration de ces données moléculaires via les méthodes d'apprentissage automatique permet aujourd'hui d'aller vers une nouvelle forme de médecine baptisée médecine de précision.

La médecine de précision vise en effet à étudier les caractéristiques très fines des patients à l'échelle de l'ADN, que constitue le profil moléculaire, dans le but de proposer aux patients un parcours de soin personnalisé et un suivi médical à différentes étapes de leur prise en charge

1. https://www.intelligence-artificielle.gouv.fr/fr/secteurs-prioritaires/l-intelligence-artificielle-au-service-de-la-sante et https://www.aiforhumanity.fr/
2. https://www.gehealthcare.fr/solutions/aidream

(diagnostic, pronostic, recherche de traitement). On parle également de prédiction de phénotypes. Généralement, cette médecine cherche à :

— diagnostiquer en détectant au plus tôt les prédispositions d'un patient à une certaine maladie. Cela peut par exemple concerner la détection d'un cancer précoce pour empêcher qu'il ne se développe plus et ainsi éviter au patient de subir des traitements lourds ;
— assurer le suivi du patient en prédisant et en anticipant la progression de la maladie ;
— déterminer le traitement le mieux adapté aux caractéristiques du patient.

Cela participe également à la découverte de nouvelles connaissances en étudiant les signatures génétiques ou indicateurs biologiques dits biomarqueurs des maladies. Le développement d'algorithmes d'apprentissage automatique précis contribue à l'émergence de cette nouvelle forme de médecine. Ces algorithmes construisent des classifieurs pour prédire les phénotypes et identifier des signatures.

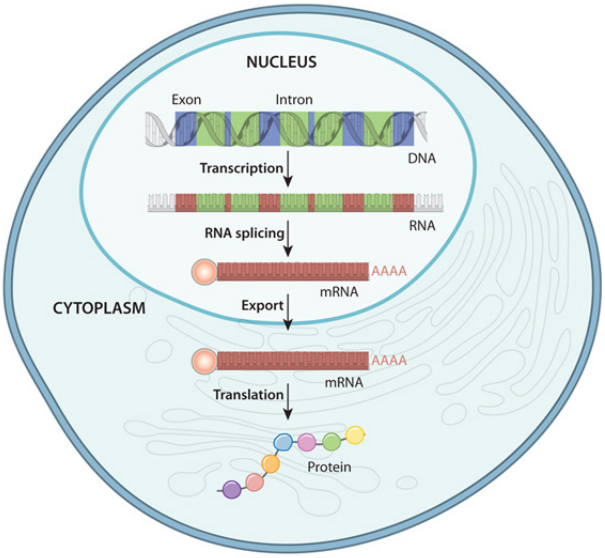


FIGURE 1.1 – ***De l'ADN à la protéine (© 2014 Nature Education).*** *L'ADN est constitué de deux brins complémentaires de nucléotides enroulés en hélice. Au cours de la transcription et de la maturation (RNA splicing en anglais), l'ARNm est produit à partir des gènes en fonction des besoins de la cellule. La protéine est le résultat de la traduction (translation en anglais) de l'ARN.*

Le profil moléculaire des patients regroupe l'ensemble des données -omiques : ADN (génomique), ARN (transcriptomique), protéines (protéomique), méthylation (épigénomique)... Rappelons que le support de l'information génétique est l'ADN stocké dans le noyau des cellules. L'expression de l'ADN se fait par l'intermédiaire d'autres molécules, l'ARN ou transcrits et les protéines, qui sont produites à quantité variable en fonction des besoins cellulaires. L'ADN est tout d'abord transcrit en ARN dit prémessager. Cet ARN prémessager devient ensuite l'ARN messager (ARNm) après avoir subi plusieurs transformations durant une phase dite de maturation. Ces deux premières phases, transcription et maturation, ont lieu dans le noyau de la cellule. L'ARNm quitte le noyau pour être traduit en protéine dans le cytoplasme cellulaire. Ces différentes étapes sont schématisées par la Figure 1.1. Seuls les gènes, qui correspondent à des fragments d'ADN dits codants (exons) à la hauteur d'1,2 %, produisent des protéines. Les parties non codantes (introns) de l'ADN (98,8 %), improprement qualifiées d'ADN poubelle dans le passé, suscitent actuellement beaucoup d'intérêt du fait de découvertes sur leur rôle biologique, entre autres, dans la régulation de l'organisation et la maintenance du génome [19]. On peut ainsi voir que l'ensemble de ces molécules interagissent entre elles.

Parmi les données omiques, les données transcriptomiques ou données d'expression de gènes (*gene expression* (GE) en anglais) jouent un rôle clé dans le développement de la médecine de

précision. Ces données sont en effet connues pour former des indicateurs de l'état cellulaire et permettre l'étude de maladies complexes telles que le cancer. Un profil d'expression génétique, qui s'exprime par une quantité d'ARN produits dans une cellule, varie en fonction des besoins de l'organisme. Nous pouvons identifier les gènes importants assurant le bon fonctionnement cellulaire, mais également repérer des mutations génétiques à l'origine de dérégulations cellulaires pouvant aboutir par exemple à un cancer. Ces gènes surexprimés ou sous-exprimés forment des biomarqueurs intéressants. Le développement de nouvelles techniques de séquençage dont RNASeq permettent de plus en plus une production en grande quantité de ces données.

Contrairement aux données d'image et de texte, les données d'expression de gènes ne sont pas structurées spatialement. Ce sont des données vectorielles proches du format des données tabulaires. Des méthodes ML supervisées aussi bien que non supervisées sont appliquées sur ces données pour extraire des biomarqueurs et étudier les phénotypes. Principalement, ce sont l'analyse en composantes principales (ACP), la machine à vecteurs de support (*support vector machine* (SVM) en anglais) et les méthodes d'ensemble dont le *boosting* et la forêt aléatoire (*random forest* (RF) en anglais) [Kou+15].

Parallèlement, l'apprentissage profond (*deep learning* (DL) en anglais) [GBC16] est une avancée majeure de l'apprentissage automatique ces dix dernières années. L'apprentissage profond s'est rapidement imposé comme un standard dans plusieurs domaines tels que la vision par ordinateur et le traitement automatique des langues (TAL) en dépassant les précédents records des méthodes ML de l'état de l'art. Le principal avantage de l'apprentissage profond vient de l'alternance de combinaisons linéaires et unités de traitement non linéaires représentées par des neurones, qui permet la construction hiérarchique de représentations latentes des données avec un niveau d'abstraction de plus en plus élevé. On qualifie alors ces modèles à base d'apprentissage profond de réseaux de neurones (*neural network* (NN) en anglais). Ces modèles ont commencé progressivement à être appliqués sur les données d'expression de gènes [Zou+18]. Le principal verrou technique de l'application de l'apprentissage profond aux données GE est principalement dû à la faible quantité d'exemples par rapport au grand nombre de variables, or l'apprentissage profond fonctionne très bien sur de grands ensembles d'apprentissage. L'apprentissage profond peut pourtant s'avérer très efficace pour découvrir des structures complexes dans les données à grande dimension que sont les données d'expression de gènes, même malgré leur manque de structure. Il est nécessaire d'adapter ces modèles pour les faire fonctionner sur ces données.

1.2 Sujet de thèse

1.2.1 Problématique

Plus largement qu'aux données d'expression, le développement de l'apprentissage profond dans le domaine de la santé se heurte à un verrou important qui est le manque d'interprétation. En effet, ces modèles, au même titre que d'autres algorithmes ML tels que les SVMs ou les méthodes d'ensembles, sont considérés comme des boîtes noires ou des modèles opaques. Cela signifie que ces modèles ne fournissent pas d'explication aux utilisateurs sur leur processus complexe de prise de décision, mais seulement leur prédiction finale. Or, si on souhaite utiliser ces modèles en pratique, l'interprétation est nécessaire, en particulier dans le domaine médical pour plusieurs raisons. D'une part, l'interprétation permet aux utilisateurs finaux (par exemple, des chercheurs, cliniciens ou patients) de comprendre les raisons qui ont permis à un algorithme de prédire un phénotype. Ceci aide aussi à vérifier qu'une prédiction est basée sur des représentations fiables des patients et non pas sur des artefacts non pertinents présents dans les données d'apprentissage. Indépendamment de l'efficacité du modèle, cela affectera les décisions de l'utilisateur final et sa confiance envers le modèle. Un utilisateur sera ainsi capable de valider ou non les décisions de ce dernier. D'autre part, un réseau de neurones performant pour la prédiction d'un certain phénotype peut avoir identifié une signature dans les données qui pourrait ouvrir sur

de nouvelles pistes de recherche et qui, autrement, n'aurait pas été détectée par l'utilisateur final. Ainsi, une question importante aujourd'hui est de rendre les algorithmes ML et en occurrence les modèles DL explicables. On parle généralement d'« IA explicable » (*eXplainable AI* (XAI) en anglais), formalisé pour la première fois par la DARPA [3] [Gun17]. Néanmoins, les premiers travaux remontent aux années 70 avec les premiers systèmes experts tels que MYCIN [SB75].

1.2.2 Objectif

L'objectif de cette thèse est de poursuivre le développement prometteur des modèles d'apprentissage profond sur les données d'expression génétique tout en s'assurant de les rendre interprétables et donc utilisables en pratique dans des contextes cliniques. Nous verrons, dans l'état de l'art actuel, que des méthodes d'interprétation ont commencé à être développées pour permettre d'interpréter a posteriori les réseaux de neurones. D'autres solutions consistent à rendre les réseaux de neurones automatiquement explicables. Cependant, ces deux approches ont été conçues essentiellement sur les données d'image et de texte. Nous avons donc cherché à développer de nouvelles méthodes d'interprétation adaptées aux données particulières d'expression de gènes qui par nature ne sont pas directement compréhensibles par l'homme, contrairement aux données d'image ou de texte. Nous proposons de ce fait de baser l'interprétation sur les connaissances du domaine pour produire des explications intelligibles aux différents utilisateurs. Trois nouvelles méthodes interprétables par construction ont été ainsi conçues pour la prédiction de phénotype sur les données d'expression. L'architecture du modèle d'apprentissage profond représente une base de connaissances biologique où chaque neurone du modèle est associé à un objet biologique et les connexions entre les neurones simulent les relations entre les objets biologiques. Chaque modèle a ses propres spécificités :

- la première méthode, Deep GONet, se base sur un perceptron multicouche contraint par une base de connaissance biologique, la *Gene Ontology* (GO), par l'intermédiaire d'un terme de régularisation adapté. Les explications des prédictions sont fournies par une méthode d'interprétation a posteriori.
- la seconde méthode, GraphGONet, tire parti à la fois d'un perceptron multicouche et d'un réseau de neurones de graphes (*graph neural network* en anglais) afin d'exploiter au maximum la richesse sémantique de la connaissance GO. Ce modèle a la capacité de rendre automatiquement des explications.
- la troisième méthode, BioHAN, ne se base plus que sur un *graph neural network* et peut facilement intégrer différentes bases de connaissances et leur sémantique. L'interprétation est facilitée par le recours aux mécanismes d'attention orientant le modèle à se concentrer sur les neurones les plus informatifs.

Les différentes méthodes développées viennent aussi enrichir l'état de l'art des approches à base d'apprentissage profond sur ces données. Ces méthodes ont été évaluées sur des jeux de données publics et ont montré leur capacité à fournir des explications intelligibles fondées sur les connaissances du domaine et à être aussi performantes que les méthodes boîtes noires de l'état de l'art.

Nous avons eu l'occasion de collaborer avec l'entreprise Oncodesign [4], pionnière dans la médecine de précision. Cette entreprise innove dans la conception d'outils à base d'IA visant à découvrir de nouvelles cibles thérapeutiques contre les cancers et les maladies graves. Ce partenariat nous a permis de pouvoir tester les méthodes développées, les adapter à leur problématique et profiter de leur expertise pour les améliorer. Nos méthodes pourront aussi leur être bénéfiques dans la découverte de nouveaux biomarqueurs.

La suite de ce manuscrit de thèse est divisée en sept chapitres. Le chapitre 2 sera consacré à

3. https://www.darpa.mil/program/explainable-artificial-intelligence
4. https://www.Oncodesign.com/fr/

des rappels sur les réseaux de neurones généralement utilisés sur les données d'expression de gènes et à une description des jeux de données publics utilisés. Dans le chapitre 3, nous reviendrons sur la terminologie autour de l'interprétation et de l'explication. Nous présenterons également l'état de l'art de l'interprétation sur les réseaux de neurones et de l'intégration de connaissances dans ces réseaux. Ensuite, les chapitres 4 à 6 présenteront les trois contributions. Les chapitres 4 et 5 sont en partie basés sur une traduction des articles publiés durant le doctorat. Enfin, le chapitre 7 sera dédié aux conclusions et perspectives de ces travaux.

Chap.

2 Préliminaires

Contenu du chapitre

2.1 Méthodes d'apprentissage profond utilisées

Dans les travaux produits durant ce doctorat, deux principaux modèles d'apprentissage profond ont été utilisés pour résoudre des tâches de classification : le perceptron multicouche (*multilayer perceptron* (MLP) en anglais) et le réseau de neurones de graphes (*graph neural network* (GNN) en anglais).

Dans les sous-sections suivantes, je décrirai ces deux types de modèles et l'auto-encodeur (AE) qui est un modèle couramment utilisé dans la littérature sur les données d'expression de gènes. Les caractéristiques essentielles de ces modèles seront données pour aider à la compréhension de cette thèse.

2.1.1 Perceptron multicouche

Un perceptron multicouche, illustré par la Fig. 2.1, est un réseau de neurones densément connecté qui possède une couche d'entrée, au moins une couches cachée et une couche de sortie retournant une prédiction dans un contexte supervisé (classification, régression). On parle généralement de couches denses. Dans le cas où il n'y aucune cache cachée, on parlera simplement de perceptron. La couche d'entrée est composée d'autant de neurones qu'il y a de variables. Chaque neurone reçoit la valeur d'expression d'une variable d'un exemple donné X. Dans le cas des données d'expression de gènes, une variable représentera un gène ou un fragment de gène. Les couches cachées sont constituées d'un nombre variable de neurones. La couche de sortie, quant à elle, contient un seul neurone dans le cas d'un problème de classification binaire ou de régression, ou un neurone par classe dans le cas d'un problème de classification multiclasses pour retourner une prédiction Y. Chaque neurone d'une couche est connecté à tous les neurones des

1. Cette image a été générée via l'outil en ligne http://alexlenail.me/NN-SVG/index.html.

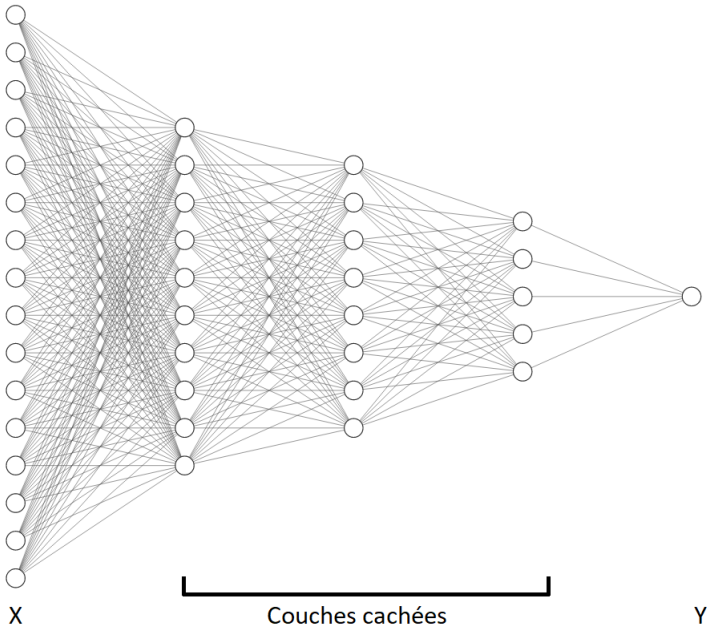


FIGURE 2.1 – ***Illustration*** [1] ***d'un perceptron multicouche possédant trois couches cachées.***

couches adjacentes et toutes les connexions sont pondérées. Il n'existe pas de connexions entre neurones d'une même couche. L'information y circule dans le sens direct de la couche d'entrée vers la couche de sortie. Ce modèle est aussi catégorisé de réseau à propagation directe ou avant (*feed forward neural network* (FFNN) en anglais). On peut finalement voir ce type de réseau de neurones comme un graphe acyclique dirigé (*directed acyclic graph* (DAG) en anglais) et valué.

L'activation du i-ème neurone de la couche cachée l s'exprime de la façon suivante : $a_i^{(l)} = f\left(\sum_{j=1}^{N_{l-1}} a_j^{(l-1)} w_{ji}^{(l)} + b_i^{(l)}\right) = f\left(z_i^{(l)}\right)$, où $w_{ji}^{(l)}$ est le poids porté par la connexion entre le j-ème neurone de la couche $(l-1)$ et le i-ème neurone de la couche l, $b_i^{(l)}$ correspond au biais du i-ème neurone de la couche l, N_{l-1} au nombre de neurones dans la couche $(l-1)$, $z_i^{(l)}$ à la somme pondérée des signaux provenant de la précédente couche $(l-1)$ vers le neurone i, et f à une fonction d'activation généralement la fonction d'unité linéaire rectifiée (en anglais *rectified linear unit function* (ReLU)) telle que $f(x) = \max(0, x)$. Pour un problème de classification binaire, le seul neurone de sortie renvoie la probabilité de prédiction de la classe positive. Celui-ci est muni de la fonction d'activation sigmoïdale telle que $a^{(L)} = \frac{1}{1-\exp(z^{(L)})}$ où L représente le nombre total de couches (cachées et de sortie). En ce qui concerne un problème de classification multiclasses, l'activation d'un neurone de sortie associé à une classe c est définie par l'application de la fonction softmax telle que $a_c^{(L)} = \frac{\exp(z_c^{(L)})}{\sum_{j=1}^{C} \exp(z_j^{(L)})}$ où C représente le nombre de classes. Notons que l'ensemble des poids et des biais forme les paramètres du MLP à optimiser durant la phase de rétropropagation décrite à la sous-section 2.1.4. Le MLP est un des modèles de réseaux de neurones le plus simple, mais qui a le désavantage de posséder beaucoup de paramètres. Généralement, c'est entre la couche d'entrée et la première couche cachée qu'il y a le plus de paramètres à apprendre. Il ne constitue pas le modèle le plus performant pour résoudre des tâches précises, notamment sur des données d'image ou de texte. Les réseaux de convolution (*convolutional neural network* (CNN) en anglais) [He+16] ou à base d'attention [Vas+17] le surpassent largement sur des tâches telles que la reconnaissance d'images ou la traduction. Par contre, les perceptrons multicouches peuvent être utilisés en combinaison de ces derniers.

2.1.2 Auto-encodeur

L'auto-encodeur, illustré par la Fig. 2.2, est un type de réseau de neurones à propagation avant, fréquemment utilisé dans le cadre de l'apprentissage non supervisé, pour réduire la dimensionnalité des données. L'objectif du réseau est d'apprendre une représentation compacte des données d'entrée par un processus d'encodage et puis, d'être capable de reconstruire à partir de celle-ci la donnée d'entrée par décodage. Le problème revient finalement à résoudre un problème de régression. L'architecture de ce modèle se rapproche d'un perceptron multicouche. Il est composé d'une couche d'entrée X, d'un encodeur $E(X)$ formé d'une à plusieurs couches cachées, d'un code Z représentant la projection la plus compacte des données, appelée également représentation latente, d'un décodeur $D(Z)$ qui est le miroir inverse de l'encodeur et d'une couche de sortie $\hat{X}$. Ainsi, la couche d'entrée et de sortie sont composées du même nombre de neurones. Généralement, les couches sont densément connectées comme dans un MLP. Il existe différentes variantes comme l'auto-encodeur débruiteur (*denoising autoencoder* (DAE) en anglais) [Vin+08] ou l'auto-encodeur variationnel (*variational autoencoder* (VAE) en anglais) [KW14]. Dans un DAE, les données d'entrée sont bruitées, par exemple par un bruit gaussien. Le modèle doit ensuite reconstruire la donnée débruitée. Cette approche est employée pour construire une représentation latente plus robuste aux bruits. Quant au VAE, il diffère de l'auto-encodeur classique dans le sens où l'objectif est d'estimer la distribution gaussienne de l'espace latent (moyenne et variance). Afin de générer $\hat{X}$, il faut donc tirer un code Z dans la distribution et le passer au décodeur.

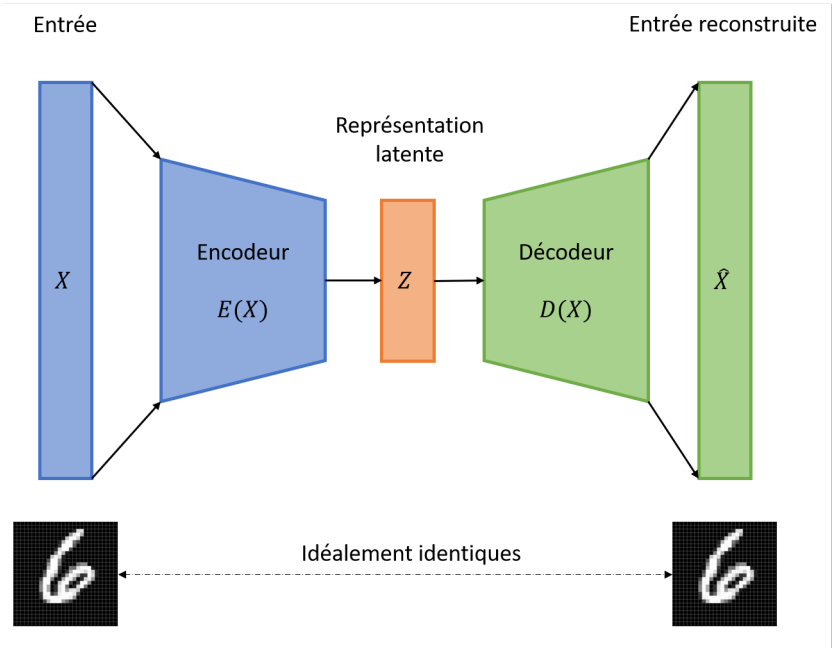


FIGURE 2.2 – ***Illustration d'un auto-encodeur simple.***

2.1.3 *Graph neural network*

Un *graph neural network* est un type de réseau de neurones opérant directement sur des structures de graphes où un graphe $\mathcal{G}$ est généralement représenté par un ensemble de nœuds $\mathcal{V}$ et d'arêtes $\mathcal{E}$ qui peuvent être orientées ou non orientées. La taille d'un graphe se mesure par son nombre de nœuds. Chaque nœud est caractérisé par un vecteur de dimension d. Une arête peut être également munie d'un vecteur de représentation. Le voisinage d'un nœud v est défini par l'ensemble $\mathcal{N}(v)$.

Un GNN peut être conçu pour résoudre des tâches de prédiction au niveau d'un nœud, d'une arête ou du graphe global. Par exemple, à l'échelle du nœud, cela peut concerner la prédiction de son étiquette, comme le sujet d'un article dans un réseau de citations tel CORA [McC+00], à partir de sa représentation vectorielle, correspondant à l'encodage binaire des mots présents dans l'article, et de l'information provenant de son voisinage, c'est-à-dire les interactions avec les autres articles. On peut faire de même à l'échelle d'une arête en prédisant si deux utilisateurs dans un réseau social ont une relation entre eux ou à l'échelle du graphe en prédisant la catégorie d'une molécule définie par un ensemble d'atomes qui forment les nœuds du graphe. Les applications sont ainsi très diverses allant du TAL aux sciences biologiques ou physiques [Wu+20] comme illustrés par la Figure 2.3. Dans la suite, nous nous intéresserons spécifiquement aux tâches de classification de graphes.

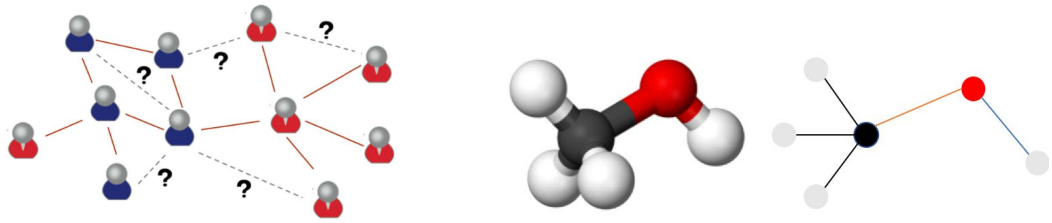


FIGURE 2.3 – ***Illustration de graphes d'application des GNNs issus de [Zho+20] (sous licence CC).*** *Un réseau social tel que Reddit [Guy+17] est présenté à gauche et une molécule comme dans le jeu de données PROTEIN [Bor+05] à droite.*

Similairement à un CNN, un GNN pour des tâches de classification de graphes est usuellement composé d'un ensemble de quatre modules qui peuvent se répéter : couche de convolution de graphes (*graph convolutional layer* en anglais), couche de réduction que l'on différencie en *pooling* et *readout*, et couche dense (MLP). Une illustration est donnée en Fig. 2.4. Pour les tâches de prédiction sur les nœuds ou les arêtes, généralement seules des couches de convolution de graphes et denses sont utilisées.

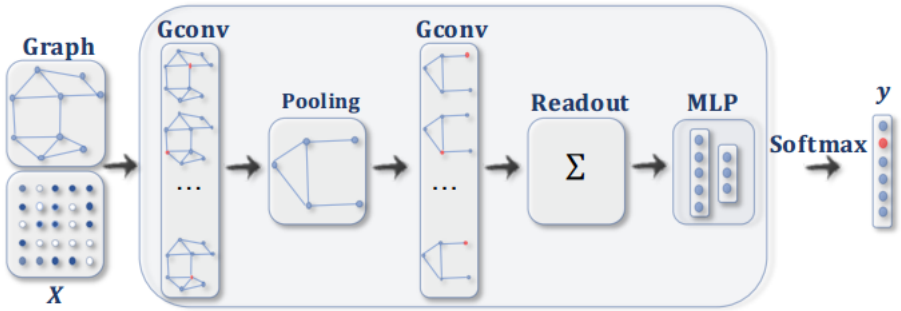


FIGURE 2.4 – ***Illustration de l'architecture d'un GNN pour une tâche de classification de graphe issue de [Wu+20] © 2020 IEEE.*** *Une couche de convolution, symbolisée par Gconv, est suivie de couches de réduction (pooling ou readout). La sortie du module de readout passe par des couches denses (MLP) pour achever la tâche de classification finale.*

Une couche l de convolution de graphe, notée GCONV, est chargée d'appliquer une loi de propagation sur chaque nœud du graphe, illustrée en Fig. 2.5. Le vecteur de représentation d'un nœud v qu'on désignera par $h_v^{(l)}$ est mis à jour à partir de la combinaison du vecteur de représentation précédent $h_v^{(l-1)}$ et de l'agrégation $h_{\mathcal{N}(v)}^{(l)}$ des informations précédentes provenant des voisins u du nœud v tel que :

$$h_{\mathcal{N}(v)}^{(l)} = \text{AGGREGATE}\big(\{h_u^{(l-1)}, \forall u \in \mathcal{N}(v)\}\big) \tag{2.1a}$$

$$h_v^{(l)} = \text{COMBINE}\big(h_v^{(l-1)}, h_{\mathcal{N}(v)}^{(l)}\big) \tag{2.1b}$$

Des exemples d'instanciation sont donnés respectivement par les équations (2.2a) et (2.2b). Le nœud v peut être inclus dans son propre voisinage.

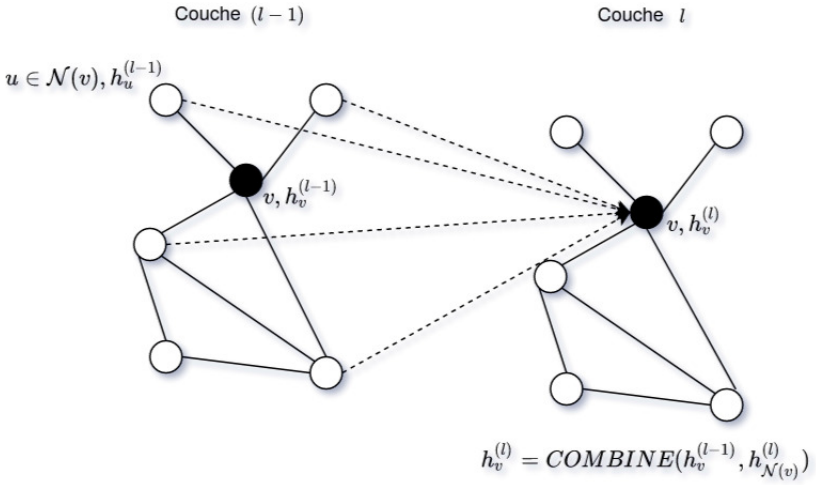


FIGURE 2.5 – ***Illustration, inspirée de [Dwi+20], de la loi de propagation d'un module de convolution de graphes appliquée entre une couche*** $(l-1)$ ***et*** l***.***

Tout comme les couches de convolution dans les CNNs, les paramètres appris sont communs aux entrées. Les deux étapes d'agrégation et de combinaison sont souvent référencées de processus de transmission de message (*message passing* en anglais) [Gil+17]. Ces étapes sont des fonctions prenant diverses formes. On distingue habituellement deux approches : l'approche spatiale basée sur la théorie des réseaux de neurones convolutifs et l'approche spectrale basée sur les théories du traitement du signal. On retrouve dans la littérature plus de modèles fondés sur l'approche spatiale. Parmi les premiers travaux portant sur cette approche, KIPF et WELLING [KW17] ont proposé le modèle *graph convolutional network* (GCN) où les équations (2.1a) et (2.1b) se déclinent en :

$$h_{\mathcal{N}(v)}^{(l)} = \sum_{u \in \mathcal{N}(v)} \frac{1}{c_{uv}} h_u^{(l-1)} W^{(l)} \tag{2.2a}$$

$$h_v^{(l)} = \sigma\Big(\frac{1}{c_{vv}} h_v^{(l-1)} W^{(l)} + h_{\mathcal{N}(v)}^{(l)}\Big) \tag{2.2b}$$

avec c_{uv} une constante de normalisation qui généralement est égale à $\sqrt{d_v d_u}$ ou d_v, ce dernier représentant le degré du nœud v soit la cardinalité du voisinage du nœud $|\mathcal{N}(v)|$. σ est une fonction non-linéaire de type ReLU et $W^{(l)} \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}}$ est la matrice de poids entre la couche $(l-1)$ et l à apprendre où $d^{(l)}$ correspond à la dimension de la représentation vectorielle d'un nœud à la couche l. On peut également voir cette formule sous la forme matricielle suivante :

$$H^{(l)} = \sigma\big(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l-1)} W^{(l)}\big) \tag{2.3}$$

où $\tilde{A} = (A + \mathbb{1}_{|\mathcal{V}|})$ est la matrice d'adjacence contenant des boucles, $\tilde{D}$ la matrice de degré correspondante, $W^{(l)}$ la matrice de poids entre la couche $(l-1)$ et l, et $H^{(l-1)}$ la représentation vectorielle des nœuds à la couche $(l-1)$. La plupart des modèles récents sont des évolutions de ce premier modèle [Guy+17 ; Vel+18 ; Xu+19]. DGP [Kam+19] est une méthode, en l'occurrence conçue spécifiquement pour les graphes orientés qui utilisent deux matrices de poids pour

distinguer l'information provenant des nœuds parents p et des nœuds fils c :

$$H^{(l)} = \sigma\Big(\tilde{D}_p^{-1}\tilde{A}_p\sigma\big(\tilde{D}_c^{-1}\tilde{A}_c H^{(l-1)}W_c^{(l)}\big)W_p^{(l)}\Big) \tag{2.4}$$

où $\tilde{D}_p^{-1}\tilde{A}_p$, $\tilde{D}_c^{-1}\tilde{A}_c$ sont respectivement les matrices d'adjacence normalisées pour les nœuds parents et les nœuds fils. Au lieu d'utiliser des matrices de poids distinctes pour traiter ces graphes orientés, nous pouvons fixer le voisinage d'un nœud comme l'ensemble des nœuds fils et appliquer les méthodes GNNs proposées sur les graphes non orientés. Néanmoins, ces méthodes ne sont généralement pas adaptées au traitement des graphes orientés particuliers que sont les DAGs. Peu de travaux traitent ce type de graphe. S'inspirant des méthodes TAL déployées sur le texte, Thost et Chen ont proposé une méthodologie combinant attention et unité récurrente de type *gated recurrent unit* (GRU) dans un processus bidirectionnel [TC21]. Ils ont aussi fait en sorte que les nœuds soient traités de manière séquentielle pour agréger l'information du voisinage la plus à jour possible. En généralisant à l couche, l'équation (2.1a) se réécrirait de la manière suivante :

$$h_{\mathcal{N}(v)}^{(l)} = \text{AGGREGATE}\big(\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\}\big) \tag{2.5}$$

Les résultats obtenus sont néanmoins peu interprétables.

La couche de réduction (*pooling* en anglais) a pour objectif de réduire la taille du graphe en extrayant les caractéristiques les plus pertinentes du graphe. On distingue trois catégories de *pooling* qui sont le *pooling* topologique, hiérarchique et global. Le *pooling* global ne tient compte que de l'information contenue dans le graphe. Il est généralement utilisé pour transformer le graphe sous forme vectorielle avant d'appliquer les couches denses dans le cadre du *readout*, similairement à ce qui est fait dans les CNNs par la couche *flatten* visant à aplatir le tenseur. Les approches classiques consistent à résumer l'information par une opération d'agrégation *sum/max/mean* qui permet de prendre la somme, la moyenne ou le maximun d'une caractéristique sur tous les nœuds pour passer d'un espace $\mathbb{R}^{|\mathcal{V}|\times d^{(l)}}$ à un espace $\mathbb{R}^{d^{(l)}}$. D'autres méthodes plus affinées ont été développées pour les GNNs telles que SortPool. SortPool classe les nœuds selon la dernière dimension de leur représentation vectorielle [Zha+18]. Les représentations vectorielles des k nœuds les mieux classés sont concaténées dans un espace vectoriel $\mathbb{R}^{k\times d^{(l)}}$ pour alimenter un MLP. Le *pooling* topologique fait appel à des algorithmes tierces de clustering tels que le partionnement en k-moyennes [Mro+18] ou encore celui de Graclus [DGK07]. Le *pooling* hiérarchique prend enfin compte à la fois de la topologie et de la signature du graphe. DiffPool appartient à cette catégorie où l'objectif est de construire des clusters de nœuds automatiquement à partir des couches de convolution [Yin+18]. Un *supra*-nœud par cluster sera ensuite défini pour le représenter et réduire ainsi la taille du graphe. L'autre approche, catégorisée sous le nom de méthodes top-k, consiste à attribuer un score d'importance aux nœuds indiquant la quantité d'information qu'ils sont capables de retenir. Ces scores sont définis à partir soit d'une projection apprise avec gPool [GJ19], soit par des coefficients d'attention appris par une couche supplémentaire de convolution avec SAGPool [LLK19] ou par un MLP avec AttPool [Hua+19]. De la même manière que SortPool, les k nœuds de score le plus élevé sont sélectionnés. Les matrices d'adjacence sont généralement mises à jour en extrayant les arêtes impliquant les nœuds choisis. Les modules de *pooling* et de *readout* s'annotent de la manière suivante : $POOL\big(h_{\mathcal{G}}^{(l)}\big)$ et $READOUT\big(h_{\mathcal{G}}^{(l)}\big)$ où $h_{\mathcal{G}}^{(l)}$ représente le signal du graphe entier à la couche l porté par les représentations vectorielles de tous les nœuds. Tout comme dans les réseaux de neurones convolutifs, il est possible d'avoir un enchaînement de plusieurs blocs de couches de convolution et de réduction avant d'appliquer le module readout et les couches denses.

2.1.4 Optimisation

Tout modèle d'apprentissage profond est avant tout un problème d'optimisation. Dans le cadre d'un problème de classification multiclasses, on formalise le problème comme un problème

de minimisation de l'entropie croisée. La fonction de coût s'écrit :

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{c=1}^{C} (-y_{i,c} \log \hat{y}_{i,c}) \tag{2.6}$$

où N et C correspondent respectivement au nombre d'exemples dans l'ensemble d'apprentissage et au nombre de classes. $y_{i,c}$ est l'indicateur de la classe positive, c.-à-d. $y_{i,c} = 1$ quand le i-ème exemple appartient à la classe c, ou 0 sinon. $\hat{y}_{i,c}$ est la probabilité calculée par le modèle que le i-ème exemple appartienne à la classe c. Pendant la phase d'inférence, nous sélectionnons la classe avec la probabilité la plus élevée pour déterminer la prédiction finale. La fonction de coût varie en fonction du type de problème à résoudre. Dans le cadre de l'auto-encodeur, l'objectif est de minimiser la différence entre la donnée d'origine et la donnée reconstruite. La fonction de coût s'écrit alors :

$$\mathcal{L} = \sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2 \tag{2.7}$$

où $\hat{x}_i$ correspond à l'exemple i reconstruit.

Peu importe le problème, l'apprentissage revient ensuite à propager l'erreur dans le sens indirect de la couche de sortie à la couche d'entrée au moyen d'une méthode de descente de gradient. On parle de rétropropagation (*backpropagation* en anglais). Les paramètres sont ainsi mis à jour selon la direction opposée au gradient. Cette mise à jour (2.8) est modulée par le pas d'apprentissage, noté ρ, permettant de converger plus ou moins rapidement vers une solution.

$$w_{\text{new}} = w_{\text{old}} - \rho \frac{\partial \mathcal{L}}{\partial w}\Big|_{w_{\text{old}}} \tag{2.8}$$

Il existe plusieurs variantes de méthodes de descente de gradient. L'une des plus connues est ADAM (*Adaptative Moment Estimation*) [KB15] avec mini-batching. Elle est la combinaison de la méthode du gradient stochastique avec moment (*Stochastic Gradient Descent (SGD) with momentum* en anglais) prenant en compte les gradients précédents, et de RMSProp (*Root Mean Square Propagation*) normalisant chaque dimension du gradient [TH12]. C'est une méthode adaptative, c'est-à-dire que le pas d'apprentissage s'ajuste au cours de l'apprentissage. Elle permet d'accélérer la convergence. Au lieu de considérer les données d'apprentissage dans leur intégralité, les données sont réparties aléatoirement au sein de N' groupes. Le temps d'une époque, la rétropropagation s'exécute sur chacun de ces groupes. Le mini-batching présente l'avantage d'accélérer l'apprentissage.

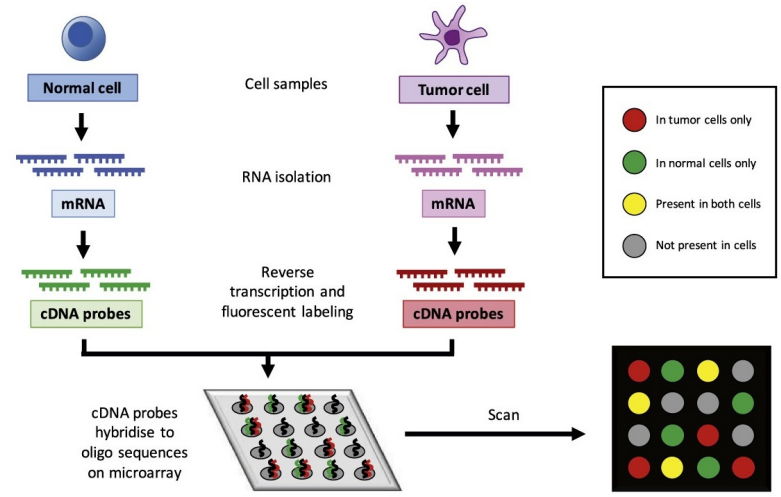
L'un des problèmes majeurs en apprentissage automatique et plus particulièrement en apprentissage profond est le surapprentissage. Le modèle est par construction très complexe et risque dans certains cas d'apprendre à parfaitement résoudre la tâche de prédiction sur les données d'apprentissage sans être capable de généraliser suffisamment à de nouvelles données. Cela est d'autant plus vrai quand la taille de la base d'apprentissage n'est pas suffisamment grande. Le *vanishing gradient* est également un autre problème qui peut apparaître à cause d'une diminution très rapide des valeurs des gradients pendant la rétropropagation entraînant l'annulation du gradient et donc l'arrêt de l'apprentissage. Dans les deux cas, on peut alors avoir recours à différentes techniques de régularisation dont la *batch normalization* visant à normaliser le signal z_i avant l'application de la fonction d'activation, le *dropout* masquant aléatoirement les neurones durant l'apprentissage selon une loi de Bernouilli, l'arrêt prématuré de l'apprentissage (*early stopping* en anglais) et la régularisation L1-L2 des poids. L'implémentation de ces différentes techniques s'accompagne d'un nombre important d'hyperparamètres à déterminer (largeur et profondeur du réseau, pas d'apprentissage, taille du batch, choix de la régularisation...). En général, pour éviter de biaiser le modèle, on utilise un ensemble de validation pour déterminer la valeur de ces hyperparamètres.

2.2 Données utilisées

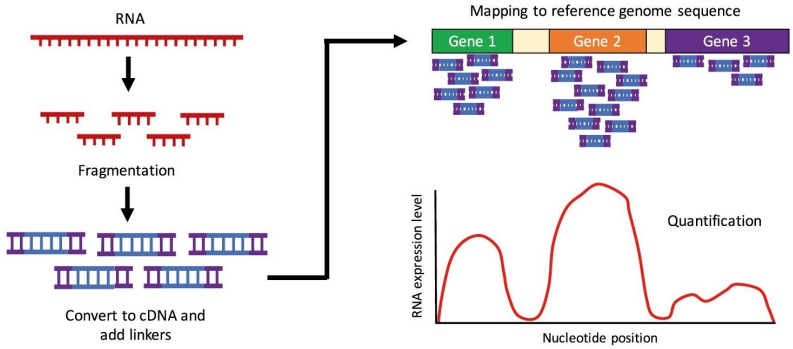
Nous allons maintenant détailler les jeux de données utilisés durant le doctorat au nombre de trois qui ont été acquis par des technologies différentes. Il existe en effet deux principales technologies de production des données d'expression de gènes (GE) : la technologie Microarray et la technologie RNASeq, la dernière étant la plus récente.

2.2.1 Description des technologies

Les technologies sont illustrées par la Figure 2.6 dont les étapes sont expliquées par les paragraphes a) et b).



(a) ***Microarray***



(b) ***RNASeq***

FIGURE 2.6 – ***Comparaison des deux technologies de mesure de l'expression génétique.*** *L'image provient de l'article [Rog+17] ©2017 Elsevier Inc. All rights reserved.*

a) Microarray

On utilise ici une puce à ADN (*microarray* en anglais) où des fragments spécifiques de gènes d'intérêt de même taille y sont fixés en rangées ordonnées dans des puits. Cette puce est ensuite introduite dans deux milieux différents : un milieu témoin (sain) et un milieu cible (pathologique). Chacun de ces milieux contient des fragments d'ARNm marqués par fluorescence : vert pour le milieu témoin et rouge pour le milieu cible. Ensuite, par un mécanisme inverse à la transcription (passage de l'ADN à l'ARN) qu'on nomme la rétrotranscription (passage de l'ARN à l'ADN), les fragments obtenus dans les différents milieux vont s'apparier par complémentarité

aux fragments d'ADN de la puce (étape **1** et **2**). On quantifie ensuite l'expression en comparant la fluorescence des deux puits et en déterminant dans quelle condition les fragments s'expriment le plus (étape **3**). Ainsi, pour un fragment donné, si le signal rouge est plus intense que le signal vert, son expression est plus importante dans le milieu cible, signe peut-être d'une mutation génétique. Au contraire, si le signal est jaune, les expressions sont identiques entre les deux milieux.

b) RNASeq

Contrairement à la technologie précédente, on fixe directement sur la puce les fragments du milieu cible d'ARNm, possiblement reconvertis en ADN, ou d'ARN non codants (étape **1** et **2**). Des nucléotides marqués sont ensuite envoyés sur la puce pour s'apparier avec les nucléotides des fragments. Les séquences ainsi obtenues seront comparées à un génome de référence grâce au séquençage à haut débit (NGS) afin d'identifier et de comptabiliser les gènes ou parties non condantes exprimés (étape **3**). Cette technologie est de plus en plus utilisée du fait qu'elle présente l'avantage de cartographier le transcriptome [2] entier contrairement à la technologie précédente. De cette manière, de nouveaux biomarqueurs peuvent être identifiés [3].

Quelle que soit la technologie utilisée, il existe différentes plateformes de production de ces données menant à de grandes variabilités dans la collecte de celles-ci. Chaque plateforme a en effet son propre protocole et peut utiliser des références génomiques différentes.

Par ailleurs, ces données ne peuvent pas être utilisées de manière brute. L'étape de prétraitement n'est pas à négliger. Les données doivent être nettoyées et normalisées. Une fois de plus, le choix de la méthode de normalisation dépendra de la technologie utilisée. D'ailleurs, il n'existe pas une unique façon de les normaliser. Certaines bases de données publiques telles que *Genotype-Tissue Expression project* (GTEx) [4] ou *The Cancer Genome Atlas* (TCGA) donnent accès à des données déjà pré-normalisées selon un certain standard. De nombreuses librairies implémentées sous R [5] permettent aussi d'automatiser cette tâche.

Chaque jeu de données va être maintenant décrit un par un. Notons que globalement ces données sont très complexes par la présence de plus de 50K variables.

2.2.2 Jeux de données E-MTAB-3732

Ce premier jeu de données, disponible en ligne sur la base de données publique ArrayExpress [6], provient d'une étude cross-expérimentale compilant les données de plus de 40 000 puces micorarray Affymetrix HGU133Plus2 [Tor+16]. Ces puces ont été produites selon différents protocoles expérimentaux (variation du lieu d'expérimentation, du stade et de la localisation du cancer) et concernent dix-sept types différents de tissus. Ces données ont été filtrées, normalisées et annotées pour former un échantillon final de 27 887 exemples dont 9450 (33,89 %) sont cancers et 18 437 (66,11 %) non-cancers. Parmi ces exemples, certains proviennent de cellules de patients (76,18 %) ou de lignées cellulaires (23,82 %), c'est-à-dire que les cellules ont été cultivées in vivo en laboratoire. Par abus de langage, on désignera par la suite un exemple de patient. Le nombre de variables est de 54 675 et représentent des fragments de gènes qui sont reliés à 23 437 gènes. Nous avons choisi de catégoriser les patients entre individus sains ou malades du fait

2. ensemble des ARN issus de la transcription du génome.
3. https://www.illumina.com/content/dam/illumina-marketing/documents/icommunity/article_2011_12_ea_rna-seq.pdf
4. https://gtexportal.org/home/index.html
5. https://www.bioconductor.org/
6. téléchargeable sous l'identifiant E-MTAB-3732 sur https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-3732/?query=p53&page=6&pagesize=25 [Kol+15]. Ce site est entretenu par l'EMBL.

que des annotations pour certains patients manquaient ou n'étaient pas cohérentes entre elles. Par exemple, le tissu de prélèvement peut parfois être inconnu, mais nous avons quand même un indicateur sur l'état normal ou tumoral de leurs cellules prélevées. La Table 2.1 récapitule la répartition des effectifs. Ces données ont été centrées-réduites et découpées en un ensemble d'apprentissage, de validation et de test avec respectivement 17 847, 4462 et 5578 exemples. Les proportions d'origine cancer/non-cancer ont été préservées dans chaque ensemble. L'ensemble d'apprentissage est utilisé pour entraîner les modèles, l'ensemble de validation sert à ajuster les hyperparamètres et enfin l'ensemble de test pour estimer les performances finales des modèles. À partir de ces données, nous nous sommes principalement intéressés au diagnostic médical (cancer/non-cancer).

Tissu	**#exemples**	**#non-cancer**	**#cancer**
abdomen	142	92	50
cerveau	869	50	819
côlon	1239	127	1112
estomac	154	31	123
foie	730	129	601
ganglion lymphatique	567	6	561
os	3525	340	3185
ovaire	573	40	533
peau	835	381	454
poumon	1415	597	818
prostate	415	65	350
pancréas	243	64	179
rein	657	257	400
sang	4283	1947	2336
sein	2171	308	1863
surrénale	83	15	68
utérus	572	26	546
Total tissu connu	18 473	4475	13 998
Total tissu inconnu	9414	4975	4439
Total	27 887	9450	18 437

Table 2.1 – ***Répartition des effectifs des données microarray E-MTAB-3732 par tissu.*** *Le symbole "#" désigne l'effectif.*

2.2.3 Jeux de données TCGA

Le deuxième jeu de données[7] est constitué de données RNASeq provenant de l'Atlas du génome du cancer[8] (*The Cancer Genome Atlas* (TCGA) en anglais) [TCW15]. Parmi les 33 types de cancer fournis par ce programme, nous avons choisi d'utiliser les types de cancer contenant plus de 350 patients. Nous avons ainsi extrait 5982 patients de onze types de cancer différents et 482 cellules saines prélevées sur les tissus étudiés, soit un total de 6464 exemples pour 12 classes. Pour chaque classe, des annotations cliniques sur les individus sont également fournies. Les 56 602 variables représentent ici à la fois des parties codantes (gènes) et non codantes du génome. La Table 2.2 ci-dessous récapitule la répartition des patients par type de cancer.

7. téléchargeable sur le portail *Genomic Data Commons* (GDC) https://gdc.cancer.gov/ géré par l'institut américain National Cancer Health (NCH)

8. https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

Type de cancer	BRCA	HNSC	KIRC	LGG	LIHC	LUAD	LUSC
#patients	1102	500	538	511	371	533	502
Fréquence	17.05 %	7.74 %	8.32 %	7.91 %	5.74 %	8.25 %	7.77 %

Type de cancer	NT	OV	PRAD	THCA	UCEC	Total
#patients	482	374	498	502	551	6464
Fréquence	7.46 %	5.79 %	7.71 %	7.77 %	8.53 %	100 %

TABLE 2.2 – ***Répartition des effectifs des données TCGA par classe.*** *NT signifie Normal Sample et désigne les exemples sains. La signification des abréviations est donnée par l'Annexe 7.2.*

Étant donné que les données acquises par la technologie RNASeq sont discrètes, les variables peuvent prendre valeur dans des intervalles très différents puisqu'en fonction de la maladie, certains gènes sont surexprimés ou sous-exprimés. Il est donc nécessaire de normaliser ces variables. Généralement, ces données sont transformées en variables continues à l'aide du format FPKM pour *Fragments Per Kilobase Per Million* [Li+17]. Cela repose sur une normalisation en utilisant le 3^{e} quartile et en prenant compte de la longueur du gène. Généralement, on applique ensuite la fonction $\log_2$ en assignant aux valeurs en dessous d'un la valeur une. Ce pipeline [9] a été appliqué aux données TCGA avant qu'elles soient finalement centrées-réduites. Tout comme pour les données précédentes, nous avons constitué un ensemble d'apprentissage, de validation et de test avec respectivement 4136, 1035 et 1293 exemples en respectant les proportions d'origine. Nous avons étudié sur ces données la prédiction du type de cancer. Néanmoins, il est possible aussi d'étudier le pronostic des patients en résolvant des tâches de survie.

2.2.4 Cohorte de données d'Oncodesign

Le jeu de données RNASeq fourni par l'entreprise Oncodesign a été constitué de manière rétrospective dans le cadre du programme OncoSNIPE, PSPC (Projets Structurants Pour la Compétitivité) en partie financé par BPI France, regroupant plusieurs partenaires industriels et cliniques. Il contient, après fusion des étiquettes et des profils d'expression, 228 patientes atteintes du cancer du sein triple négatifs. Il y a 130 patientes non répondeuses (NR) et 98 répondeuses (R) à leur première ligne de traitement, soit une légère disproportion de 0,57 pour la classe majoritaire NR. Le problème auquel on s'intéressera est un problème de classification binaire (répondeur/non-répondeur) visant à déterminer la réponse à un traitement. Les données ont été normalisées suivant le même protocole que précédemment. Vu que nous disposions de peu de patients par rapport aux nombres de variables, nous avons dû passer par une validation croisée en 10 blocs.

2.3 Apprentissage profond sur les données transcriptomiques

Les architectures utilisées dans la littérature sur les données GE sont principalement des auto-encodeurs et des perceptrons multicouches [DM19]. L'objectif de l'auto-encodeur est d'apprendre une représentation compacte des profils d'expression génétique dans une couche latente qui aurait capturé une information biologique utile pour la tâche de prédiction finale réalisée par un classificateur tel qu'un SVM ou un MLP [TCF17 ; DGH17 ; WG18]. L'avantage est que pour apprendre ce type de modèle, il n'est pas nécessaire de posséder une grande base de données étiquetées. Par exemple, FAKOOR et al. [Fak+13] réduisent la dimension des données GE en passant d'abord par une ACP ; cette première réduction est augmentée avec l'expression de certains

9. https://docs.gdc.cancer.gov/Data/Bioinformatics_Pipelines/Expression_mRNA_Pipeline/

gènes d'origine sélectionnés aléatoirement qui passeront dans un auto-encodeur épars (*sparse autoencoder* (SAE) en anglais) ou un *stacked autoencoder* où des auto-encodeurs successifs sont construits prenant compte de l'information passée. Une régression softmax est enfin appliquée sur la représentation compressée obtenue pour résoudre une tâche de détection de cancer ou de prédiction du type de cancer. L'auto-encodeur appris peut aussi servir de pré-initialisation d'un MLP par apprentissage par transfert. Hanczar et al. [Han+18] utilisent, quant à eux, un MLP pour des tâches similaires en pré-entraînant chaque couche avec un auto-encodeur débruiteur.

Les perceptrons multicouches peuvent être utilisés pour prédire directement les phénotypes sans réduction de dimension [Che+20a ; Han+20 ; Kat+18]. Chen et al. [Che+20a] proposent, par exemple, de régulariser un MLP en ajoutant une erreur liée au clustering sur la dernière couche cachée. L'objectif est de maximiser la marge entre chaque classe dans l'espace latent défini dans cette couche cachée afin de fournir une meilleure prédiction du cancer en couche de sortie. Katzman et al. [Kat+18] ont, dans leur cas, développé un MLP adapté pour résoudre des problèmes de survie. Ils ont notamment remplacé la couche de sortie par une couche Cox à risque proportionnel. Quelques travaux [Mos+20 ; KMR20] sont basés sur des réseaux convolutionnels qui sont généralement appliqués sur les données d'images matricielles [KSH12]. Contrairement à ces dernières, les données GE n'ont pas de spatialité et sont arrangées donc aléatoirement pour former une matrice 2D pour qu'un CNN puisse être appliqué. Certains travaux ont essayé d'intégrer des informations externes pour identifier une structure dans l'expression des gènes qu'un CNN ou un GNN peuvent exploiter. Par exemple, quelques travaux se basent sur les GNNs où un patient est représenté par un graphe de gènes de coexpression [Ram+20 ; Xin+21].

Malgré des premiers résultats prometteurs, l'apprentissage profond n'a pas autant percé que pour l'image et le texte dans la prédiction de phénotypes à partir des données d'expression de gènes, en raison de la taille souvent réduite des ensembles d'apprentissage disponibles. Néanmoins, dans un futur proche, il est fort probable que l'apprentissage profond jouera un rôle majeur pour résoudre ces problèmes du fait d'une production de plus en plus croissante de ces données [HBZ22].

Chap.

3 État de l'art

Contenu du chapitre

3.1 Terminologie

Il convient tout d'abord de définir le vocabulaire qui sera utilisé tout le long de cette thèse. Qu'entendons-nous par *explication* (ou *explicabilité*) et *interprétation* (ou *interprétabilité*) ? La 9e édition du dictionnaire de l'Académie française donne la définition suivante au mot interprétation « explication du sens qu'on peut donner à un texte » et par extension le fait d'« attribuer une signification à un phénomène, un fait », parfois obscur ou ambigu. On verra dans explication le fait de « faire comprendre quelque chose, donner les causes, les raisons d'un fait ». Explication et interprétation sont liées. Néanmoins, on réserve souvent en français le mot interpréter pour rendre compte d'une opinion personnelle sur un résultat (champ de la subjectivité), tandis qu'expliquer est plutôt relatif à une justification factuelle de ce résultat (champ de l'objectivité). En anglais, on retrouve également cette dualité. En effet, selon le dictionnaire de Cambridge, le mot interprétation est à la fois synonyme d'explication et d'opinion, et à la définition d'explication, est ajouté le caractère de *compréhensibilité* des raisons fournies. Dans les travaux en apprentissage automatique (ML) à ce sujet, Montavon et al. [MSM18] décrivent l'interprétation comme le fait de « projeter un concept abstrait (par exemple, une classe prédite) dans un domaine que l'homme peut comprendre » et l'explication comme le fait de « fournir des raisons compréhensibles à l'homme, par exemple, l'ensemble de variables qui a le plus contribué à une prédiction donnée ». On y retrouve la définition anglaise de l'explication, mais celle propre à l'interprétation est redéfinie de manière à englober celle de l'explication. Au lieu d'employer les termes d'explication ou interprétation, certains usent des termes d'explicabilité et d'interprétabilité. L'interprétabilité au sens de Doshi-Velez et Kim [DK18] signifie la « capacité d'expliquer ou de présenter en termes compréhensibles à un être humain ». Cette définition s'avère très proche de la définition d'explication et est de plus en plus utilisée aux dépens du terme d'interprétation. Gilpin et al. [Gil+18] ajoutent également que « l'objectif de l'interprétabilité est de décrire le fonctionnement interne du modèle de manière compréhensible ». Le terme d'explicabilité, quant à lui, est souvent utilisé comme un synonyme d'interprétabilité dans des documents de politique générale, tandis que la communauté ML préfère le terme d'interprétabilité [Bea+]. Certains,

au contraire, emploient le terme d'interprétabilité (resp. explicabilité) comme synonyme d'interprétation (resp. explication). L'accès au fonctionnement interne d'un modèle repose également sur la notion de *transparence*[1]. Ce terme présente également diverses significations. Un système transparent, selon [Lip18], doit satisfaire des critères de compréhension à différents niveaux du modèle, par exemple, au niveau des composantes individuelles comme les paramètres ou encore au niveau de l'algorithme d'optimisation. Pour d'autres, il doit inclure des détails sur comment le système d'IA a été développé, entraîné et déployé [Bea+]. Cette notion peut ainsi englober celle d'interprétabilité et d'explicabilité [CH19]. Or, on décrit souvent, même en présence du code source, qu'un réseau de neurones est opaque, car son fonctionnement n'est pas compréhensible dans des termes à portée de tous. Nous conserverons cette notion d'*opacité* par la suite pour qualifier entre autres les modèles d'apprentissage profond dont l'architecture n'est pas facilement appréhendable. Enfin, le terme d'interprétation peut être parfois réservé à l'interprétation globale du modèle et le terme d'explication à l'interprétation locale d'une prédiction, rejoignant la définition donnée par Montavon et al. [MSM18]. Nous pourrons ainsi voir l'interprétation du modèle comme une agrégation des explications. Cette distinction n'est pas systématiquement faite et les deux termes sont employés de manière interchangeable. Par la suite, nous nous contenterons d'employer les termes d'interprétation et explication selon cette dernière distinction du fait de l'ambivalence des différents termes évoqués. De notre point de vue, l'objectif de l'interprétation au sens large est de détecter les éléments des données et du modèle sur lesquels se basent les prédictions de manière globale, voire locale dans le cadre d'une explication. À travers les différentes terminologies proposées, nous avons pu voir que la dimension d'*intelligibilité* de l'interprétation est essentielle ; certes, une explication peut être fournie en révélant les variables les plus explicatives, mais elle n'a de sens que si ces variables sont compréhensibles. Or, ceci n'est pas le cas sur les données moléculaires, contrairement aux données textuelles ou d'images. Bien évidemment, le degré d'intelligibilité dépendra des connaissances de l'utilisateur final.

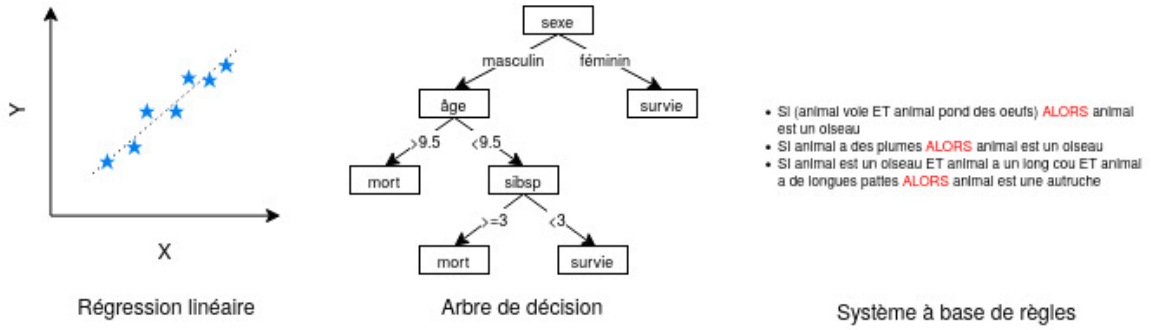


Figure 3.1 – ***Illustration des trois méthodes ML interprétables par essence les plus populaires.*** *L'exemple de l'arbre de décision est tiré de l'exemple Wikipédia sur le jeu de données Titanic et celui du système à base de règles d'un exemple sur la classification animale du cours de l'association ABORD*[2]*. La variable sibsp dans l'arbre de décision signifie nombre de frères et sœurs/conjoints à bord.*

Dans la suite de cette thèse, nous nous concentrerons sur l'interprétation des réseaux de neurones du fait que ce soient les modèles utilisés dans nos approches. Nous proposons une catégorisation des différentes approches pour interpréter les réseaux de neurones qui est résumée par la Table 3.1. Avant de débuter, il est important de souligner l'existence de modèles dits

1. Cette notion est ambigüe. Si le code source d'un modèle ML est fourni et qu'il est suffisamment clair et commenté, ce modèle peut être jugé transparent puisqu'on peut avoir accès aux différentes étapes du processus d'apprentissage et d'inférence. Notons que la mise en ligne du code (*open source* en anglais) et du jeu de données est de plus en plus exigée par les journaux ou conférences en ML traduisant la volonté d'aller de plus en plus vers une science ouverte. On dénombre de nombreuses initiatives à ce sujet, notamment avec le second Plan National pour la science ouverte en France.

2. http://www.abord-ch.org/cours/ia/chap1.htm

interprétables par essence ou par nature, parfois qualifiés de *glass or white box* en anglais. Illustrés dans la Fig. 3.1, ces modèles sont entre autres les modèles linéaires, les arbres de décision, ou encore les systèmes à base de règles. Une liste plus exhaustive peut être trouvée dans le livre de Molnar [Mol20]. Grâce à une faible complexité et un fonctionnement transparent et compréhensible, il est facile de retracer le processus de décision du modèle et de déterminer les variables de plus grand impact sur la décision. Par exemple, dans un modèle linéaire, les poids du modèle indiquent directement l'importance de chaque variable. Ces modèles sont interprétables globalement et localement. Néanmoins, pour être suffisamment à portée de main, ces modèles doivent être relativement petits. Cette complexité spatiale peut être mesurée via des indicateurs comme le nombre de paramètres non nuls dans le cas d'un modèle linéaire, la profondeur et la largeur de l'arbre dans le cas d'un arbre de décision, et le nombre de prémisses dans le cas d'un système à base de règles. Ces indicateurs représentent le nombre d'unités qu'un utilisateur doit être capable d'assimiler. Ainsi, on préfèrera des modèles linéaires parcimonieux (*sparse* en anglais) comme la méthode Lasso visant à mettre à zéro certains coefficients du modèle linéaire. Enfin, les méthodes de réduction de dimensions peuvent également impacter la compréhension de ces modèles. Il faudra prendre en compte que les variables d'entrée sont potentiellement transformées et que vis-à-vis de celles-ci, les unités de ces modèles n'ont plus la même significativité [Lip18].
Dans l'état de l'art actuel, il existe deux approches pour interpréter les réseaux neurones :

- l'approche dite a posteriori (*post hoc* en anglais), la plus utilisée, utilisant une seconde méthode dédiée à l'interprétation, une fois que le réseau de neurones a été appris. Ces méthodes seront désignées de méthodes d'interprétation.
- l'approche dite *self-explaining*, que l'on pourrait traduire par auto-explicable, visant à créer des modèles qui se rapprochent des modèles interprétables par essence, comme sont les méthodes ML classiques décrites ci-dessus.

À la catégorisation offerte ci-dessous, on ajoutera une seconde dimension liée à l'interprétation du modèle (global) et celle de la prédiction (local). La classification des méthodes selon ces deux dimensions est souvent utilisée. Que ce soit une approche a posteriori/*self-explaining* ou une interprétation locale/globale, il peut exister des similitudes dans les méthodologies utilisées. C'est ce que nous avons souhaité mettre en évidence dans le tableau suivant pour faciliter la lecture de la suite. Nous pouvons ainsi regrouper les méthodes dans des familles selon les caractéristiques communes qu'elles partagent. Certaines méthodes sont, de cette sorte, basées sur la notion de prototype ou de concept. Nous entendons par concept le fait d'identifier des motifs (ou *patterns*) de haut niveau sémantique sur lesquels le modèle se base pour faire ses prédictions. Un prototype, au contraire, est une instance réelle ou synthétique qui pourrait représenter toutes les autres ou celles d'une classe précise et ainsi aider à l'interprétation. Certaines fois, ces deux notions sont confondues et sont employées en synonyme. Les méthodes de substitution, quant à elles, reposent sur le principe d'avoir recours à une méthode subsidiaire, généralement une méthode interprétable par essence, pour approximer localement ou globalement le réseau de neurones. Concernant les méthodes d'attribution, elles calculent la contribution relative de chaque neurone du réseau à l'aide d'un score dit de pertinence. Les méthodes contrefactuelles et contrastives formalisent un nouveau problème d'optimisation, tandis que les méthodes à base d'attention contraignent le réseau à se focaliser sur certains éléments des données, mais importants. Nous verrons que certaines méthodes peuvent se retrouver dans plusieurs familles comme la méthode LIME [RSG16]. Elles peuvent enfin recourir à l'emploi de scores (attention, contribution) ou de perturbations qui consistent à introduire des bruits dans les données, souvent imperceptibles à l'œil nu, ou parfois dans le modèle, et à en évaluer le changement sur les prédictions.
Notons que la classification peut être également organisée selon le type d'explication [Zha+21b], le type de données d'entrée [Bod+21], ou encore en fonction de la spécialisation de la méthode à un type de boîte noire [Bar+20]. Les interprétations ou explications fournies peuvent aussi prendre différentes formes telles qu'une image, un texte, une liste...

Familles	Méthodes	*self-explaining*/post hoc	locale	globale	perturbation	score
Recherche sémantique (prototype/concept)	AM [SVZ14 ; Ngu+16 ; Zho+15]	post hoc		✓	✓	
	Dissection [Bau+17], TCAV [Kim+18]	post hoc	✓	✓		✓
	ExMatchina [Jey+20],[Car00]	post hoc	✓			
	ConceptBottleneck [Koh+20a], SENN [AJ18], [Li+18]	*self-explaining*	✓	✓		✓
Méthodes de substitution	DeepRED [ZLJ16], modèle hybride [WL21]	post hoc	✓	✓		
	LIME [RSG16], DeepSHAP [LL17]	post hoc	✓		✓	✓
	INVASE [YJS19], FLINT [PMd21]	*self-explaining*	✓	✓		✓
Méthodes d'attribution	Sensibilité et de décomposition [SVZ14 ; Bac+15 ; SGK17 ; STY17]	post hoc	✓			✓
	Méthodes de perturbations [ZF14 ; Zho+15]	post hoc	✓		✓	✓
Méthodes contrefactuelles/contrastives	[VDH20]	post hoc	✓		✓	
Méthodes à base d'attention	Transformer [Vas+17]	*self-explaining*	✓			✓

TABLE 3.1 – *Classification des différentes méthodologies d'interprétation des réseaux de neurones selon les critères suivants : approche self-explaining/post hoc, interprétation locale/globale, usage de perturbations ou de scores particuliers. Les méthodes ont été regroupées en famille du fait qu'elles partagent des caractéristiques communes.*

3.2 Interprétation a posteriori

Dans le cadre de cette approche, l'interprétation n'est pas fournie par le modèle. Il est nécessaire d'avoir recours à une méthode tierce pour pouvoir produire une interprétation. Cette interprétation peut se faire au niveau du modèle (global) ou au niveau de la prédiction (local). Nous allons présenter les deux façons d'aborder l'interprétation.

3.2.1 Interprétation globale du modèle

Les deux familles de méthodes a posteriori permettant d'interpréter globalement un modèle boîte noire, que sont les méthodes de recherche sémantique et les méthodes de substitution, vont être présentées dans les sous-sections suivantes.

a) Méthodes de recherche sémantique

Dans cette première famille, on distinguera deux catégories : les méthodes à base de prototypes et celles à base de concepts. Dans le premier cas, on cherche à déterminer un ou plusieurs représentants par classe qui pourront servir comme support général d'interprétation. Dans le second, on cherche à identifier des caractéristiques sémantiques de haut niveau et globales dans les données qui peuvent être associées à des concepts compréhensibles par l'homme et favoriser l'interprétation.

i) Méthode de maximisation de l'activation à base de prototypes L'objectif de cette méthode (*activation maximization* (AM) en anglais) est de trouver un prototype par classe, une sorte d'entrée favorite, qui permet de maximiser la prédiction ou l'activation d'un neurone spécifique dans des couches arbitraires. La détermination du prototype passe par la résolution d'un nouveau problème d'optimisation. Ce problème peut être entre autres résolu par la descente du gradient [SVZ14]. Les réseaux antagonistes génératifs (*generative adversarial network* (GAN) en anglais) peuvent être aussi utilisés pour produire ces prototypes [Ngu+16]. S'inspirant des méthodes à base de perturbations qui seront présentées en détail plus loin, Zhou et al. [Zho+15] ont proposé une version d'AM simplifiée qui consiste à déterminer un prototype par l'injection de bruits en entrée sous la forme d'une suite d'occlusions. L'opération a lieu de manière itérative sur les K-premières entrées pour qui l'élément ciblé présente déjà la plus forte activation. Les régions des entrées candidates qui activent le plus l'élément ciblé sont ainsi repérées, permettant d'affiner le prototype. Le prototype final résulte de la moyenne des régions repérées. Cette méthode dans la description proposée ne permet pas d'expliquer individuellement une prédiction. Dans le cadre des deux premières méthodologies, le désavantage majeur est de devoir solutionner un nouveau problème d'optimisation qui peut être difficile à résoudre dans le cadre de l'apprentissage d'un réseau génératif. De plus, le réseau génératif a l'inconvénient d'être lui-même un modèle boîte noire. Enfin, les régions les plus activées ne correspondent pas forcément toutes aux éléments sur lesquels le modèle base ses prédictions. L'inconvénient des méthodes basées sur les perturbations sera discutée dans la sous-section c)iii).

ii) Méthodes à base de concepts Cette catégorie regroupe essentiellement la méthode de dissection de réseau [Bau+17] et la méthode de vecteurs d'activation de concepts [Kim+18]. La méthode de *dissection de réseau* [Bau+17] évalue la présence de concepts prédéfinis dans les représentations latentes apprises par un réseau de neurones. Elle a été originalement évaluée sur des images dans le cas d'un problème de classification. L'objectif est de vérifier si le réseau se base sur ces concepts pour produire ses décisions. Ces concepts peuvent à la fois représenter des caractéristiques de bas niveau comme des couleurs et de haut niveau comme des objets. L'évaluation passe par une mesure de l'intersection sur l'union (IoU) entre les représentations d'origine d'une catégorie de concepts et les régions des images activées par chacun des neurones.

Ces régions résultent de l'interpolation des cartes d'activations des neurones à la dimension des concepts. La carte d'activations d'un neurone est construite à partir des activations unitaires dépassant un certain seuil sur chacune des images d'un échantillon. Notons que l'ensemble des catégories de concept est évalué avec ces cartes d'activations, peu importe la classe prédite. Dans le même alignement, les méthodes déconvolutionnelles et d'inversion de réseau [ZF14] cherchent à identifier des motifs interprétables à partir des cartes d'activations d'images, respectivement, en projetant celles-ci dans la distribution des images d'origine ou en reconstruisant ces dernières.

La méthode de *vecteurs d'activation de concept* [Kim+18], plus connue sous l'abréviation TCAV pour *Testing with Concept Activation Vectors* en anglais, mesure de manière différente la sensibilité des représentations latentes avec des concepts prédéfinis. Pour ce faire, un modèle linéaire interprétable est construit sur les représentations latentes des concepts prédéfinis et celles des données d'entrée. Le vecteur orthogonal à la frontière de décision est récupéré pour pouvoir calculer, pour un ensemble de données d'entrée issues de la même classe, un score de sensibilité conceptuelle sur la dérivée directionnelle. Ce score mesure l'alignement de la représentation latente des concepts avec celle des données. Ceci permet de vérifier le degré d'importance de la présence d'un concept prédéfini par l'utilisateur dans le résultat de la classification, par exemple, la sensibilité d'une prédiction de "zèbre" à la présence de rayures. On n'évaluera pas forcément tous les concepts avec toutes les classes.

TCAV a été premièrement évaluée sur des données d'images où il est plus facile de déterminer les concepts recherchés. Néanmoins, il est possible d'appliquer cette méthode sur d'autres types de données. La méthode de dissection de réseau ainsi que les deux autres méthodes citées restent assez propres au type de réseau convolutionnel et aux données d'image utilisées. Elles demanderont un effort plus important pour être adaptées à d'autres types de réseaux de neurones et de données. Ensuite, cette catégorie de méthodes a été présentée sous l'angle de vue de l'interprétation globale du modèle, mais elles peuvent tout à fait être utilisées pour fournir une explication de la prédiction d'une entrée donnée. L'un des inconvénients est qu'il faut définir à l'avance ces concepts et que la notion de concept perceptible est peu exploitable sur des données d'entrée comme les données d'expression de gènes. Il est néanmoins possible de faire appel à la connaissance pour organiser l'information extraite sur ces gènes et révéler des concepts sous-jacents. Certaines fois, les résultats peuvent être biaisés par l'intervention de l'homme sur les choix des concepts à utiliser. Il se peut aussi que le modèle n'ayant pas appris directement sur les concepts génèrent des scores d'alignement faibles.

b) Méthodes de substitution

L'objectif de cette seconde famille est d'approximer le réseau de neurones entier par un modèle interprétable par essence. En général, l'approximation servira plus à fournir une explication de la prédiction qu'une interprétation globale. Comme cela reste une approximation, ces modèles de substitution sont des représentations imparfaites des relations apprises par le réseau de neurones. Il peut donc avoir un écart entre les informations extraites fournies par ces modèles et ce que le réseau de neurones a réellement appris. Malgré que ces modèles soient interprétables par essence, leur complexité spatiale peut impacter leur interprétation. Par exemple, si un arbre de décision est trop profond, il sera difficile à l'homme de le comprendre. Il faudra alors trouver un compromis entre la fidélité avec le modèle boîte noire et la complexité du modèle de substitution. L'objectif est d'avoir un modèle approximatif à la fois fiable et peu complexe, ces deux objectifs pouvant s'avérer antagonistes. En général, une fois défini, le modèle de substitution ne remplacera pas le réseau de neurones pour effectuer la tâche de prédiction. Il ne sera utilisé que pour la partie interprétation. Zilke et al. [ZLJ16] proposent une méthode appelée *Deep neural network Rule Extraction via Decision tree induction* (DeepRED) qui est capable de simplifier le réseau de neurones profond en un arbre de décision parcimonieux par suppression des variables non utiles via l'algorithme *C4.5* [Sal94]. Néanmoins, la génération de l'arbre de décision requiert beaucoup

de mémoire et de temps de calcul. Un exemple de systèmes à base de règles est l'algorithme KT (*KnowledgeTron*) [Fu94 ; Hai16] qui cherche à imiter le comportement de chacun des neurones. Selon une approche dite décompositionnelle, l'algorithme procède couche par couche pour extraire des règles si-alors par établissement de seuils. Les deux méthodes présentées ne prennent en compte que l'information provenant de la couche d'entrée et non pas des couches cachées, pouvant aboutir à des modèles peu fiables. De plus, elles passent difficilement à l'échelle dans le cas du traitement de réseaux de neurones profonds, car trop gourmandes en temps de calcul et mémoire. On parle de scalabilité limitée [Gil+18]. Une dernière solution est la mise en place de modèles hybrides [WL21] où le modèle de substitution est presque *auto-suffisant* sur une sous-région des données. Une fois le modèle de substitution défini, son fonctionnement est le suivant. Si l'exemple de test peut être directement traité par celui-ci, la décision ne sera rendue que par lui avec une fidélité de 100 % si la décision avait été prise par le modèle boîte noire. Au contraire, si aucune décision ne peut être prise, le modèle boîte noire prendra le relais et dans ce cas précis, une autre méthode d'interprétation a posteriori devra être utilisée pour donner une explication à la décision rendue. Il existe donc un dilemme entre les décisions rendues par le modèle de substitution et celles de la boîte noire. Cette méthode se situe au chevauchement entre les méthodes d'interprétations locales et globales et peut être appliquée sur n'importe quelle boîte noire. Néanmoins, on se retrouve confronté aux mêmes limites que les solutions citées précédemment (scalabilité, temps de calcul). Pour une littérature plus complète, le lecteur peut se référer à la récente revue des méthodes d'interprétation de Zhang et al. [Zha+21b].

3.2.2 Interprétation locale du modèle (Explication)

Tout d'abord, comme expliqué, certaines catégories de méthodes introduites précédemment, à savoir les méthodes à base de concepts et les méthodes de substitution, peuvent être utilisées pour interpréter localement un réseau de neurones. Dans cette section, ne seront présentées que les familles de méthodes qui ne servent qu'à expliquer une prédiction, soient les méthodes à base de prototypes, une seconde famille de méthodes de substitution, les méthodes d'attribution et les explications contrefactuelles et contrastives.

a) Méthodes à base de prototypes / Explication par l'exemple

L'objectif de cette famille est de trouver un ou plusieurs représentants (ou prototypes) du jeu d'apprentissage les plus proches de l'exemple de test à expliquer [Jey+20]. Cela peut passer par un calcul de distance entre les prototypes et les exemples de test. Les exemples similaires devraient avoir les mêmes prototypes. Le choix des prototypes reste à définir par le concepteur ou l'utilisateur. Un exemple de méthode est l'explication basée sur l'étude de cas [Car00]. Les exemples de test passent au travers du réseau et les cartes d'activations sont extraites. Elles sont ensuite comparées avec celles des données d'entraînement de mêmes prédictions pour identifier parmi ces derniers les prototypes.

b) Méthodes de substitution

Cette famille, proche de celle précédemment présentée sur l'interprétation globale, se base également sur l'usage des méthodes ML interprétables par essence pour produire ici uniquement une explication. On recense la méthode LIME (*Local Interpretable Model-agnostic Explanations*) [RSG16] et SHAP (*SHapley Additive exPlanations*) [Sha53]. LIME consiste à générer de nouvelles données dans le voisinage de la donnée réelle à expliquer [RSG16]. De là, un modèle de substitution local, un modèle linéaire entre autres, est construit à partir des prédictions de la boîte noire sur les données générées. L'objectif est d'obtenir à la fois un modèle parcimonieux et de minimiser la différence de prédiction entre le modèle de substitution et la boîte noire, pondérée par une mesure de similarité entre les données réelles et la donnée à expliquer. Cela aboutit à un

modèle interprétable localement à partir duquel nous pouvons extraire l'importance de chaque variable. Cette approche est relativement coûteuse, car elle nécessite de créer un modèle linéaire pour chaque décision à expliquer.

SHAP, quant à elle, repose sur la théorie des jeux [Sha53] où l'objectif est de déterminer l'importance de chacune des variables. La valeur de Shapley d'une variable est définie comme la contribution à la différence entre la valeur prédite par le modèle et la prédiction moyenne. Les valeurs de Shapley calculées peuvent être perçues comme les poids d'un modèle linéaire. Il n'est cependant pas nécessaire de le construire. Il existe différentes variantes en fonction des types de boîtes noires. DeepSHAP [LL17] est la version adaptée aux réseaux de neurones qui est combinaison des valeurs de Shapley et d'une des méthodes d'attribution détaillée ci-dessous, DeepLIFT [SGK17].

c) Méthodes d'attribution

L'objectif de ces méthodes est d'identifier les éléments discriminants d'un réseau de neurones. Pour ce faire, la contribution d'une variable d'entrée ou d'un neurone est calculée relativement à la prédiction. Cette catégorie regroupe entre autres l'analyse de sensibilité et de saillance, les méthodes de décomposition et de perturbations, ainsi que LIME et SHAP. Originalement, ces méthodes ont été appliquées aux données d'image. De ce fait, la plupart des explications produites par ces méthodes prennent la forme d'une carte de saillance (*saliency map* en anglais) mettant en évidence la contribution de chaque pixel d'une image donnée. Un gradient de couleur indique souvent le degré de contribution (ou d'importance) de chaque pixel. Il oscille entre valeurs positives (contributions positives) et négatives (contributions négatives). Étant donné une classe prédite, on entend par contribution positive que le pixel ou le neurone participent à la prédiction de cette classe alors qu'en cas de contribution négative, l'élément a tendance à dévier le signal de la classe prédite. Les cartes de saillance ont l'avantage d'offrir une forme d'explication visuellement accessible qu'un humain peut facilement interpréter sans prérequis de connaissances. La Fig. 3.2 résume cette méthodologie. On recense trois catégories de méthodes qui sont l'analyse de sensibilité et de saillance, les méthodes de décomposition et les méthodes de perturbation.

i) Analyse de sensibilité et de saillance L'analyse de sensibilité (*Sensitivity Analysis* (SA) en anglais) permet d'analyser la sensibilité (ou réaction positive) du modèle par rapport à ces unités (variables d'entrée, neurones). Elle consiste à calculer la valeur absolue de la dérivée partielle d'une sortie par rapport à une unité [SVZ14]. Il suffit de rétropropager le signal comme effectué durant l'apprentissage. Le résultat reflète davantage les variations des unités du modèle qui ont permis d'aboutir à la prédiction. Une variable d'entrée est par exemple de forte contribution si en la perturbant le moins possible par rapport aux autres variables, la prédiction s'en retrouve changée. Notons qu'en prenant la valeur absolue, nous n'avons pas d'indication sur la contribution positive ou négative de chacune des unités. Cette métrique est approximative et reste assez sensible aux bruits. Une extension de cette approche est la méthode *Gradients×Entrées* (*Gradients×Inputs* (GI) en anglais) qui peut être vue comme le produit de la sensibilité et de la proéminence ou saillance d'une unité. En effet, la dérivée partielle (sensibilité) est multipliée par la valeur de l'unité étudiée (saillance) [SVZ14]. Cette méthode a été initialement proposée pour améliorer la netteté des cartes de saillance issues de l'analyse de sensibilité et informer sur le signe de la contribution.

ii) Méthodes de décomposition Cette catégorie repose sur une décomposition du signal de sortie jusqu'à l'entrée, c'est-à-dire que la contribution d'un neurone d'une couche dépend de la contribution des neurones de la couche supérieure avec lesquels il interagit. Cette catégorie regroupe plusieurs méthodes dont *Layerwise Relevance Propagation* (LRP) [Bac+15 ; MSM18],

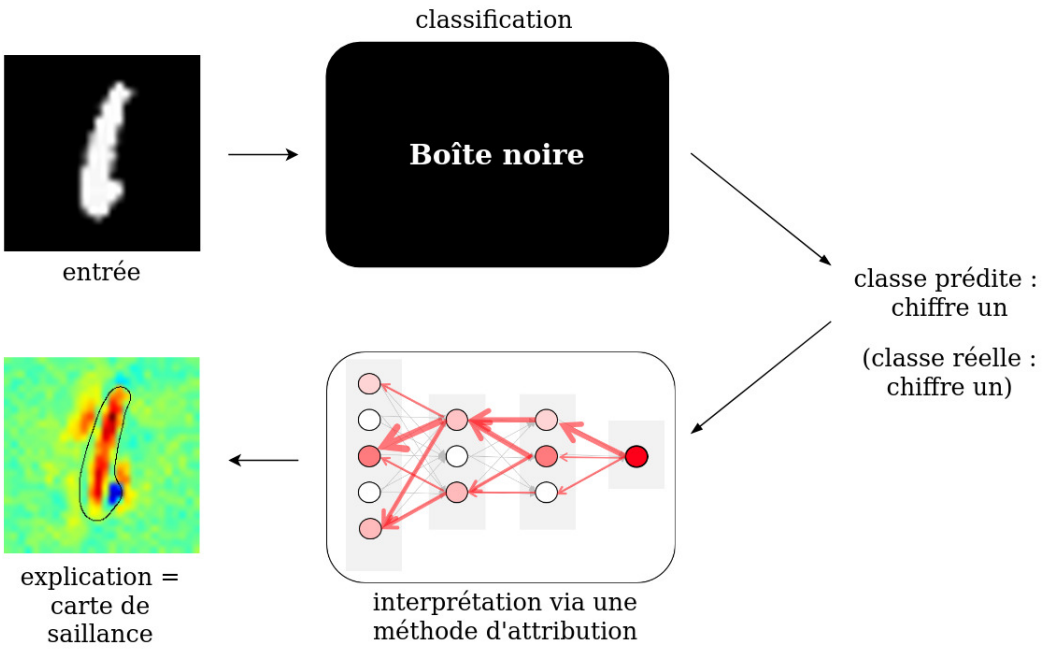


FIGURE 3.2 – ***Illustration du fonctionnement d'une méthode d'attribution.*** *L'exemple issu de MNIST a été généré à partir de la plateforme Explainable AI Demos de l'Institut Fraunhofer pour les télécommunications. Les neurones de forte contribution sont coloriés en rouge. La carte de saillance obtenue met en évidence les régions de l'image d'entrée qui ont une contribution positive en rouge et négative en bleu.*

Deep Learning Important FeaTures (DeepLIFT) [SGK17], *Integrated-Gradients* (IG) [STY17]. Dans LRP, le score de contribution représente la proportion du signal de sortie passant par le neurone et ses connexions sortantes. Il prend donc en compte la valeur d'activation du neurone et les poids des connexions, et est calculé selon une règle de propagation. Il existe différentes règles LRP-$\{z,\epsilon,\alpha\beta,\gamma,\ldots\}$ qui sont plus ou moins adaptées en fonction du type de modèle d'apprentissage profond. Il est possible de combiner ces règles dans un même modèle, on parle de stratégie composite [Koh+20b ; Mon+19]. À chaque couche du réseau, une règle est choisie en fonction de sa profondeur. DeepLIFT et IG sont un mixte entre les méthodes de décomposition et de perturbation. Cela passe par la définition d'une référence qui servira ensuite à calculer une différence entre la prédiction à expliquer et la prédiction associée à cette référence. Pour les images, la référence correspond à une image où les pixels sont nuls. Cependant, ce choix n'est pas aussi simple lorsqu'il s'agit de traiter des types de données non intelligibles. Une étude comparée des méthodes citées peut être trouvée dans [Anc+19]. Cette étude a notamment montré que DeepLIFT et LRP peuvent être considérées, sous certaines conditions, comme des méthodes à base de gradients au même titre que les méthodes SA, GI et IG. Il existe de nombreuses variantes à l'intersection entre les méthodes d'analyse de sensibilité et de saillance et celles de décomposition telles que Grad-CAM [Sel+17], SmoothGrad [Smi+17], PatternAttribution [Kin+18]...

iii) Méthode de perturbations Cette catégorie vise à perturber les variables d'une entrée donnée et d'en observer les changements qui en résultent [ZF14 ; PDS18 ; FPV19 ; DG17]. Si cela produit un changement de prédiction, c'est un indicateur que les caractéristiques perturbées ont une certaine importance pour la prédiction. Différents types de perturbation existent : masquage aléatoire/nul, bruit quelconque comme un recadrage, un passage à l'échelle de gris dans le cas d'une image. ZHOU et al. [Zho+15] ont proposé, par exemple pour une image donnée, de déterminer à partir d'une suite d'occlusions l'information minimale permettant d'activer un élément ciblé tel qu'un neurone d'une couche de sortie ou d'une couche intermédiaire. Certaines

méthodes d'évaluation sont parfois proposées telles que la *suppression* (*deletion* en anglais) pour mesurer la baisse de performance en perturbant progressivement les données d'entrée [PDS18]. Notons que les méthodes LIME et SHAP sont aussi catégorisées de méthode de perturbations. Il est reproché à cette catégorie de manquer de fondements théoriques. De plus, les perturbations choisies sont parfois perçues comme peu significatives [FPV19]. Cette méthodologie peut être employée en combinaison avec une méthode de décomposition pour valider les unités de plus grande contribution [SWM18]. En effet, si la prédiction se base sur ces unités, en les perturbant, celle-ci devrait s'en trouver également affectée.

Certaines études ont montré que les méthodes d'attribution peuvent s'avérer peu robustes aux bruits de faible ampleur et aux attaques adverses [Kin+19 ; Sla+20]. De plus, il peut arriver que les cartes de saillance produites par des réseaux de neurones aléatoires soient similaires à des réseaux entraînés [Ade+18 ; SGL20]. Il est donc nécessaire de passer par une étape de vérification des cartes obtenues en ayant recours par exemple à une étude comparative des cartes par différentes méthodes dans différents scénarios d'ablation. Cette évaluation n'est pas aussi simple puisqu'on n'a pas accès à la vérité terrain. Tout un volet de la recherche sur l'interprétation des réseaux de neurones repose également sur son évaluation. Dans le cas des cartes de saillance, l'utilisateur va être souvent amené à juger la pertinence des cartes. Il va entre autres vérifier la concordance des régions mises en valeur avec sa propre perception du monde. Ces cartes peuvent parfois être difficiles à déchiffrer. Prenons l'exemple de la classification d'images de voiture, imaginons que la région capturée met en évidence les contours de la voiture avec en arrière-plan une route. Nous ne sommes pas sûrs que le modèle base réellement son attention sur la voiture ou sur l'arrière-plan qui pourrait laisser penser qu'il s'agit d'une voiture. Il peut également arriver que pour une même classe, on obtienne des cartes de saillance différentes.

d) Explications contrefactuelles et contrastives

Dans le même alignement que les méthodes de perturbations, la famille de méthodes contrefactuelles et contrastives vise à identifier des liens de causalité entre les entrées et les sorties. L'explication contrefactuelle fournit un retour d'information du type « que se passerait-il si un point de données d'entrée x subissait un changement donné, la sortie d'un modèle ML serait-elle y' au lieu de y ». L'objectif est d'identifier les changements dans l'entrée qui contredisent les observations, mais qui vont changer la valeur de la sortie vers une sortie prédéfinie. Un certain nombre de critères doivent être respectés dont la validité du changement (la donnée doit rester assez proche de la donnée d'origine), la parcimonie (peu de variables doivent être changées) et la causalité (un changement dans une variable peut entraîner un changement sur d'autres variables) [VDH20]. Ces critères sont formulés sous la forme d'un problème d'optimisation. L'explication contrastive est assez proche de l'explication contrefactuelle. On s'intéresse plutôt à un retour d'information du type plutôt que : « pourquoi un exemple x a été prédit y et pas y' ? ». Le but est d'identifier un échantillon x' proche de x qui produit y' au lieu de y. Par exemple, une question contrefactuelle serait : « l'étudiant aurait-il échoué à l'examen s'il avait été plus attentif en cours ? » et contrastive : « qu'est-ce qui a fait la différence entre l'étudiant qui a échoué et les étudiants qui ont réussi l'examen ? ».

Notons la présence de méthodes agnostiques comme LIME qui peuvent être utilisées sur d'autres boîtes noires comme les SVMs ou les forêts aléatoires, et celle de méthodes spécifiques ne pouvant être appliquées que sur des réseaux de neurones, voire sur certains types de réseaux et pour certains types de données (cf. : les méthodes de dissection de réseau décrites p. 23). Le choix de telle ou telle famille ou catégorie de méthodes va beaucoup dépendre du but recherché. Par ailleurs, pour obtenir une interprétation globale, on pourrait procéder à l'interprétation d'un échantillon d'exemples, par exemple issus d'une même classe, pour pouvoir obtenir une

vision globale. L'autre approche qui permettrait de s'affranchir d'avoir une méthode algorithmique secondaire à paramétrer serait de se digérer vers la conception de modèles d'apprentissage profond *self-explaining* [Rud19 ; Elt20]. Toutefois, l'approche a posteriori reste intéressante pour interpréter les modèles déjà entraînés et ont l'avantage de ne pas déteriorer les performances des boîtes noires. Quelques études ont essayé d'interpréter a posteriori les modèles entraînés sur les données moléculaires en ayant notamment recours aux méthodes d'attribution telles que la méthode décompositionnelle DeepLIFT [FFB19] ou une méthode de perturbation [Ahn+18] pour déterminer respectivement les petits ARN ou les gènes de plus grande importance. Fiosina et al. [FFB19] ; Ahn et al. [Ahn+18] ont poursuivi par une analyse statistique plus approfondie pour rendre accessibles ces scores aux utilisateurs. Ils ont ainsi réalisé du clustering pour dégager des groupes de patients. Néanmoins, l'application de ces méthodes et leur validation sur les données moléculaires reste limitée si elles ne sont pas complétées par la connaissance biologique.

3.3 Vers des modèles *self-explaining*

L'objectif principal de cette nouvelle classe de modèles est de s'affranchir des méthodes a posteriori. En effet, ces modèles sont capables de fournir une forme d'explication de leurs décisions et peuvent être interprétables globalement. Sous certaines conditions, ils pourraient être qualifiés d'interprétables par essence au même titre que les méthodes ML classiques précédemment citées telles que les arbres de décision. Dans cette classe, on comptabilise trois familles : l'apprentissage simultané d'un interpréteur et d'un prédicteur, les méthodes de recherche sémantique et enfin celles à base d'attention.

a) Méthodes de substitution - apprentissage simultané d'un interpréteur et d'un prédicteur

L'objectif ici est de construire un *interpréteur* et un *prédicteur* en même temps. L'interpréteur et le prédicteur peuvent dans certains cas coopérer ou apprendre l'un de l'autre. INVASE (*INstance-wise VAriable SElection*) [YJS19] est l'une des premières méthodes de cette famille implémentant trois réseaux de neurones dans un scénario acteur-critique. Un premier réseau *baseline* correspond au modèle boîte noire qui fait ses prédictions sur l'ensemble brut des variables d'entrée. Un second réseau (qualifié de *sélecteur* ou d'*acteur*) est chargé d'identifier un sous-ensemble de caractéristiques personnalisé pour chaque entrée, soit une sélection par instance, sur lequel un troisième réseau, le *prédicteur* ou *critique*, doit être capable de faire une prédiction aussi fiable que celle de la boîte noire. Le sélecteur sera récompensé si les prédictions convergent. Les trois réseaux sont ainsi optimisés en même temps. Une fois l'apprentissage terminé, la baseline n'est plus nécessaire. La prédiction aura l'avantage d'être donnée directement par le prédicteur avec comme support d'explication le sous-ensemble défini par le sélecteur, ce dernier correspondant à l'interpréteur dans notre définition. Notons que les auteurs de cet article ont défini cette méthode comme post hoc. Parekh et al. [PMd21] ont, quant à eux, proposé FLINT (*Framework to Learn With INTerpretation*) une méthode où seulement deux réseaux, le prédicteur (semblable à la baseline dans le modèle précédent) et l'interpréteur, sont appris en même temps. L'interpréteur a pour objectif d'apprendre automatiquement des attributs de haut niveau sémantique, semblables à des concepts, à partir de représentations de certaines couches latentes. Un modèle linéaire suivi d'une activation softmax est appliqué par-dessus. Un score de relevance est ainsi calculé pour chaque attribut pour évaluer son impact final sur la prédiction. Le nombre et la position des couches à sélectionner, ainsi que le nombre d'attributs sont des hyperparamètres à déterminer. Le nombre d'attributs ne doit pas être trop grand pour permettre à l'utilisateur de mieux appréhender les explications. Cette méthode peut aussi être utilisée sur un prédicteur déjà appris.

Une interprétation globale et locale peut être menée avec ces deux méthodes en observant, à

l'échelle d'une prédiction individuelle ou d'une classe, les variables ou attributs de plus grande contribution. L'inconvénient principal de cette famille est que le modèle est rendu explicable grâce à un second réseau, lui-même opaque. Quelques désaccords entre l'interpréteur et le prédicteur peuvent aussi apparaître.

b) Rechercher sémantique

Contrairement aux méthodes a posteriori à base de prototypes ou de concepts décrites en sous-section ii), l'objectif de cette famille est d'intégrer directement ces éléments sémantiques dans l'apprentissage du modèle : soit en faisant correspondre un neurone à un concept (méthode dite associative), soit en alliant la sortie d'une couche avec des vecteurs de prototypes (méthode de similarité et d'alignement). Dans ces cas précis, les termes de concept et de prototype sont presque employés comme synonyme.

i) Méthode associative (neurone, concept) Dans cette catégorie, on peut citer deux exemples : SENN (*Self-Explaining Neural Network*) [AJ18] et *Concept Bottleneck* [Koh+20a]. FLINT [PMd21] peut d'une certaine manière aussi rentrer dans cette catégorie. La méthode SENN [AJ18] construit un modèle interprétable qui se comporte localement comme un modèle linéaire. Le modèle est divisé en deux branches. La première branche vise à apprendre les concepts qui représenteront une combinaison non-linéaire des variables d'entrée. La seconde branche est censée déterminer l'importance de chaque concept de haut niveau sémantique. Un modèle linéaire est ensuite établi à partir d'une combinaison linéaire des concepts appris et de leur importance relative. Tout comme dans FLINT, le nombre de concepts représente un hyperparamètre du modèle à fixer. Dans le cadre de la méthode *Concept Bottleneck* [Koh+20a], le réseau est divisé en deux parties. La première partie vise à projeter les données dans un espace de concepts prédéfini et la seconde partie à faire une prédiction à partir de cette nouvelle représentation. Cette représentation peut être rectifiée par les utilisateurs. L'accès à une banque de concepts suffisamment nettoyée et annotée peut être parfois difficile à obtenir, les auteurs ont donc proposé différentes manières d'optimiser le modèle en présence de ces concepts. De plus, ils ont montré qu'il suffit d'un nombre de concepts bien plus petit que la dimension d'entrée des données. FLINT et SENN ont, toutes deux, l'avantage d'automatiquement construire des concepts, contrairement à *Concept Bottleneck*.

ii) Méthode de similarité et d'alignement L'objectif de cette catégorie est de réguler le réseau de neurones en contraignant l'une de ses couches cachées à se rapprocher de prototypes définis. Cela passe par une mesure de similarité au niveau d'une couche dite de prototypes entre des prototypes définis et une représentation latente d'une donnée d'entrée fournie par un encodeur [Li+18]. Cette mesure représentera ensuite l'entrée d'un classifieur. L'ensemble de prototypes est échantillonné à partir d'un hypercube pendant l'apprentissage. La fonction de coût comprend l'ajustement des prototypes qui ont les contraintes suivantes. Chaque prototype doit être aussi proche que possible d'au moins un des exemples d'apprentissage dans l'espace latent. Chaque représentation latente doit être aussi proche que possible de l'un des prototypes. Cette stratégie a connu plusieurs évolutions. Une première évolution a été de mesurer une similarité sur la base de *patch*, c'est-à-dire qu'au lieu de considérer les représentations latentes et les prototypes de manière globale, on travaille sur des parties [Che+19]. Chaque classe dispose d'un jeu de *patchs* de prototypes qui représenteront des *patchs* du jeu d'entraînement. La seconde évolution est de s'affranchir d'une mesure de similarité en passant par une normalisation des représentations latentes à partir de prototypes définis depuis une banque de prototypes [CBR20].

De même que pour la famille précédente, une interprétation locale et globale peut être effectuée. Tout comme les méthodes a posteriori, ces méthodes ont été seulement appliquées à

des données d'image et dans la plupart des cas, il est question de réseaux convolutionnels, de sorte que les prototypes définis qui soient générés ou définis par une tierce personne soient compréhensibles par les utilisateurs. Comme discuté précédemment, il n'est pas facile d'identifier les concepts sous-jacents sur des données de base difficilement compréhensibles naturellement. Nous pouvons nous demander également si ces méthodes restent aussi performantes que les réseaux de neurones opaques. Les auteurs de *Concept Botteleneck* [Koh+20a] ont fait une étude comparative avec une méthode d'apprentissage profond classique (ResNet-19 ou Inception-v3) et ont montré que leur solution est aussi performante. Cependant, cela n'est pas tout le temps démontré. L'avantage des méthodes a posteriori est qu'elles ne dégradent pas les performances des réseaux de neurones opaques. La question de l'autosuffisance de ces méthodes peut être également posée [Yeh+20].

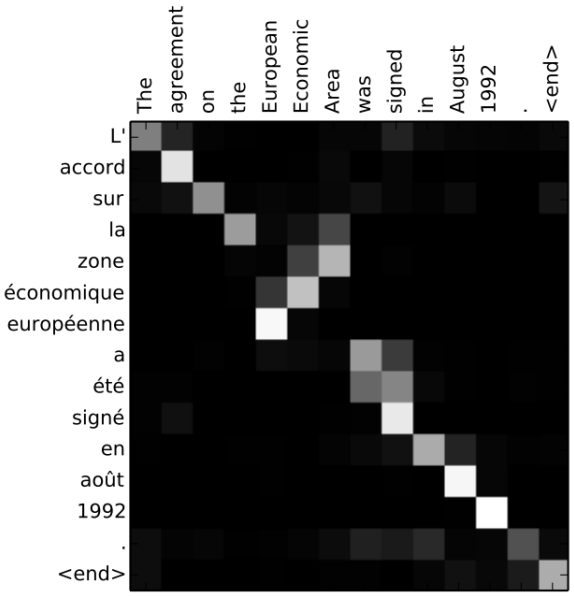


FIGURE 3.3 – ***Illustration d'une carte d'attention*** [3] ***issue de la traduction d'une phrase en français vers l'anglais (sous licence CC).*** *Le gradient de couleur (noir au blanc) indique le degré d'attention entre chaque paire de mots, du mot original vers le mot traduit. Plus la couleur est claire, plus l'attention est élevée.*

c) Méthode à base d'attention

La technique dite de l'attention a tout d'abord vu le jour dans les modèles d'apprentissage profond dédiés au TAL [Vas+17]. Cette technique a permis de considérablement améliorer les résultats dans ce domaine. Lors d'une traduction d'une séquence de mots, cette technique vise à permettre au modèle de se concentrer sur les parties les plus pertinentes de la séquence d'entrée. Cela passe par un calcul d'un score d'attention fondé sur l'alignement (ou similarité) entre chaque élément en cours de traduction et l'ensemble des mots de la séquence d'entrée. En continuité, le mécanisme d'auto-attention ou intra-attention cherche à calculer un score d'attention entre chaque mot d'une même séquence. On peut identifier pour un mot traduit dans une carte d'attention comme représentée par la Fig. 3.3, les mots de la séquence d'entrée auxquels le modèle a accordé son attention. Ces cartes, au même titre que les cartes de salliance, sont des indicateurs d'interprétation qui peuvent aider à comprendre la prédiction d'un modèle. Un vif débat autour de cette question anime la communauté XAI [JW19 ; WP19]. Certains chercheurs mettent en effet en garde sur l'usage des coefficients d'attention pour expliquer les prédictions.

3. provenant du site : https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/.

Jain et Wallace [JW19] ont montré qu'il n'y avait pas forcément de corrélation entre les coefficients d'attention et les scores d'attribution. L'inspection de ces coefficients d'attention pour l'interprétation doit donc être réalisée avec minutie. Les auteurs de cette étude proposent ainsi un ensemble de recommandations dans le cas où le mécanisme d'attention peut être utilisé pour interpréter localement un modèle.

L'ensemble des méthodes de cette approche *self-explaining* constitue une piste intéressante vers la conception de modèles d'apprentissage profond intrinsèquement interprétables. Une méthode peut être plus privilégiée que d'autres en fonction des attentes de l'utilisateur. Par exemple, si l'utilisateur, tel qu'un expert, est plus intéressé de connaître les caractéristiques de bas niveau qui ont menées aux prédictions, la méthode FLINT ou une à base d'attention peuvent être employées. Au contraire, un utilisateur lambda, sera possiblement plus intéressé par le recours aux méthodes basées sur les concepts. On peut citer quelques adaptations de ces méthodologies sur les données moléculaires comme SENN [Qui+20], des modèles à base d'attention [ML22]. Notons que les auteurs d'INVASE [YJS19] présentent dans leur article des expérimentations sur des données cliniques réelles de patients. Toutefois, dans certaines circonstances, ces approches ne sont pas autosuffisantes. Au même titre que les méthodes a posteriori pour une interprétation locale, les méthodes d'apprentissage simultané d'un interpréteur et d'un prédicteur et celles basées sur l'attention ont une utilisation limitée du fait que les scores produits s'interprètent difficilement lorsqu'il s'agit d'appliquer les méthodes sur des données par nature peu intelligibles. Pour rendre ces scores plus accessibles, il faudrait compléter par la connaissance. Cette connaissance peut être ainsi exploitée a posteriori en ayant recours par exemple à des tests statistiques complémentaires. Les méthodes à base de concepts définis sont, quant à elles, plus exploitables pour rendre les modèles interprétables par essence. Les concepts restent à définir pour une application sur les données moléculaires. Il existe une multitude de bases de connaissances biologiques et médicales qui peuvent servir de source de concepts. Dans ce cas précis, les connaissances seront utilisées a priori et pourront aider par exemple à définir l'architecture des réseaux de neurones.

3.4 Intégration des connaissances

Les connaissances biologiques sur les gènes et leurs dérivés (transcrits, protéines) regroupent différents types d'information et caractérisent de manière générale ces molécules ou de manière plus spécifique, par exemple par rapport à un type de pathologie. Ces connaissances s'organisent autour de graphes d'annotation sous forme de réseaux ou d'ontologies pouvant être orientés ou non orientés. Parmi les connaissances les plus connues, nous pouvons lister les suivantes :

— les fonctions issues de l'ontologie des gènes, la *Gene Ontology* (GO) [Con04], qui offrent une description des gènes et des produits géniques selon trois angles : les composantes cellulaires dans lesquelles ils agissent (abrégé GO-CC), les fonctions moléculaires réalisées (abrégé GO-MF), et enfin les processus biologiques dans lesquels ils sont impliqués (abrégé GO-BP). Chaque sous-ontologie est un graphe acyclique orienté (*directed acyclic graph* (DAG) en anglais - voir Fig. 3.4a) ;

— les réseaux métaboliques qui décrivent un ensemble de voies métaboliques (suites séquentielles de réactions biochimiques) ayant lieu dans une cellule. Il existe notamment KEGG (*Kyoto Encylclopedia of Genes and Genomes*) [KG00] et Reactome [Fab+18] où dans ce dernier les voies métaboliques sont hiérarchisées dans un DAG (voir Fig. 3.4b) ;

— les réseaux non orientés d'interactions protéines-protéines (abrégées IPP) décrivant l'ensemble des interactions physiques et fonctionnelles entre protéines. Il existe de nombreuses bases dont STRING *(Search Tool for the Retrieval of Interacting Genes/Proteins)* [Sne+00] (voir Fig. 3.4c) ;

— les réseaux de régulation ou de coexpression des gènes, les réseaux de signalisation, etc.

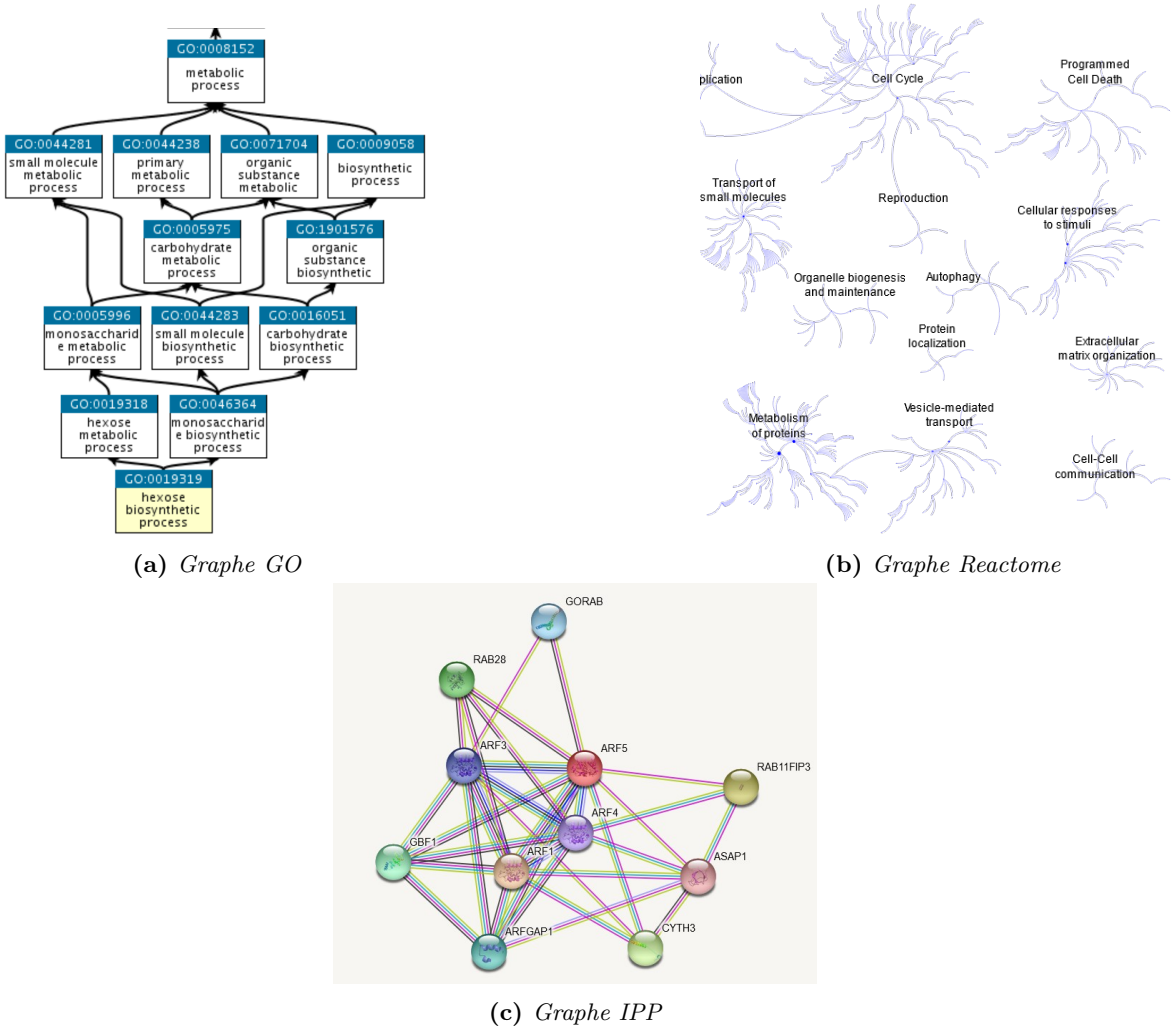
Du fait de leur popularité auprès de la communauté bio-informatique, ces connaissances ont l'avantage d'être bien maintenues.

3.4.1 Intégration des connaissances a posteriori

Dans la littérature sur l'interprétation des réseaux de neurones appliqués aux données moléculaires, une première possibilité est le recours aux connaissances après l'entraînement (a posteriori). L'objectif est d'identifier les variables d'entrée importantes via une méthode d'interprétation post hoc comme les méthodes d'attribution et puis d'effectuer un test d'enrichissement statistique sur cet ensemble afin d'obtenir la liste des concepts biologiques les plus surreprésentés. Quelques travaux ont été réalisés dans ce sens [LH18 ; Han+20]. Lyu et Haque [LH18] ont appliqué GuidedCAM sur les sorties d'un réseau convolutionnel appliqué aux données d'expression de gènes. Un test d'enrichissement par les voies métaboliques a ensuite été réalisé sur les gènes de plus grande contribution. Hanczar et al. [Han+20] ont, quant à eux, eu recours à LRP pour déterminer d'une part les neurones les plus importants et, d'autre part, les variables d'entrée contribuant le plus à l'activation de ces neurones. Ceci a permis de caractériser biologiquement chacune des couches après réalisation d'un test d'enrichissement en voies métaboliques et en fonctions biologiques sur chaque ensemble de variables obtenues. La connaissance, dans ce cas, ne contraint pas le modèle. Il existe des incertitudes sur les concepts biologiques récupérés, car ils dépendent du choix de l'ensemble des neurones jugés de plus grande importance retournés par la méthode post hoc et des hyperparamètres utilisés dans l'ajustement de la méthode d'interprétation utilisée. Une solution est alors d'exploiter les connaissances au moment de l'apprentissage du réseau de neurones. Cependant, la solution d'intégrer les connaissances a posteriori reste utile lorsque le modèle a déjà été appris.

3.4.2 Intégration des connaissances a priori

Nous pouvons intégrer les connaissances dans l'apprentissage de deux manières : soit en imposant une contrainte à l'architecture du réseau de neurones à propagation avant, soit en manipulant directement les connaissances par des méthodes dédiées aux graphes comme les



(a) *Graphe GO* **(b)** *Graphe Reactome*

(c) *Graphe IPP*

FIGURE 3.4 – ***Aperçu de la structure de trois graphes de connaissances parmi les plus connus.***

graph neural network (GNN). Dans les deux cas, les concepts biologiques sont associés à des neurones, compréhensibles par l'utilisateur. On parlera de méthodes basées sur les connaissances pouvant aider à rendre les modèles *self-explaining*.

i) Réseaux de neurones à propagation avant contraints Ces réseaux sont souvent référencés de réseaux de neurones visibles (*Visible Neural Network* en anglais) [Yu+18a] ou de modèles interprétables par construction (*interpretable by design* en anglais). Dans cette première alternative, l'architecture du réseau de neurones, tel qu'un MLP, se base sur la structure d'un graphe de connaissances où chaque neurone des couches cachées correspond à un concept biologique et chaque connexion entre deux neurones à une relation du graphe de connaissances. En pratique, pour faire cette correspondance, des masques binaires encodant les relations de l'ontologie utilisée sont généralement appliqués sur les matrices de poids de réseaux denses pour faire correspondre les neurones avec les concepts biologiques sous-jacents et contraindre l'information à ne circuler que par les connexions représentant des relations du graphe de connaissances (voir Fig. 3.5). On désigne souvent ces connexions masquées de partielles. Les réseaux de neurones listés ci-dessous disposent en général d'une à plusieurs couches de concepts biologiques en fonction de la connaissance utilisée. Les ontologies Reactome et GO étant basées sur un DAG offrent la possibilité de construire différentes couches de connaissances (voir Fig. 3.6). Les concepts qui

sont représentés dans la première couche cachée sont choisis de sorte qu'ils soient connectés à un minimum de variables d'entrée.

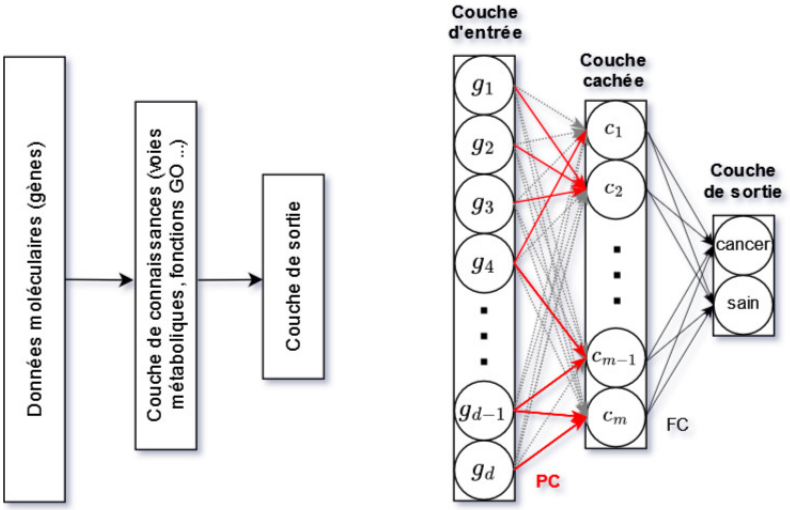


FIGURE 3.5 – ***Exemple d'intégration d'une couche de connaissances dans un réseau de neurones à propagation avant de type MLP.*** *g_i désigne le i-ème gène de la couche d'entrée, c_j désigne un j-ème concept biologique tel qu'un régulateur [Kan+17], une voie métabolique [Hao+18 ; Gau+20], ou une fonction biologique [PWS19], d et m correspondent respectivement au nombre de gènes et nombre de concepts biologiques. Les connexions entre la couche d'entrée et la couche cachée sont partielles (désignées par PC). Les connexions en rouge sont associées à des relations de la connaissance et celles grisâtres et en pointillées sont celles qui sont masquées, n'étant pas associées à des relations de la connaissance. La couche cachée et la couche de sortie sont totalement connectées (désignées par FC).*

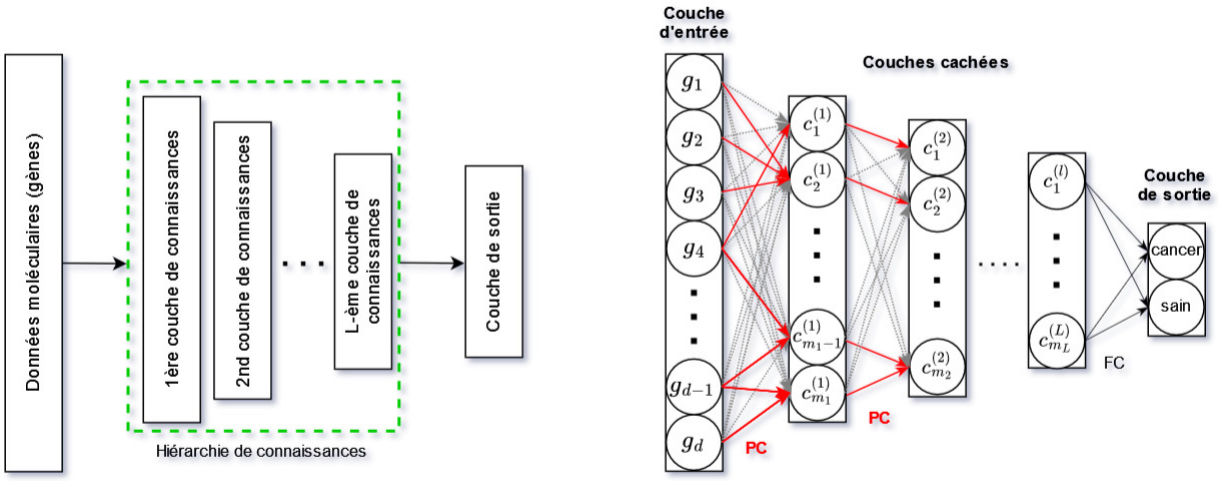


FIGURE 3.6 – ***Exemple d'intégration d'une hiérarchie de connaissances telles que Reactome [Elm+21 ; LL21] ou GO [Hua+21].*** *$c_j^{(l)}$ désigne le j-ème concept biologique de niveau l de la hiérarchie de connaissances. $m^{(l)}$ désigne le nombre de concepts biologiques dans le l-ème niveau sélectionné.*

Par exemple, KANG et al. [Kan+17] ont utilisé le graphe de régulation des gènes pour déterminer les connexions entre les neurones de la couche d'entrée représentant les gènes et les neurones de la première couche cachée de leur réseau GRRANN (*Gene Regulatory network-based Regularized Artifical Neural Network*) représentant des régulateurs tels que des protéines, miR-

NAs ou des composantes qui régulent les gènes. Cette couche cachée est ensuite directement reliée à un neurone de sortie pour résoudre un problème de classification. Dans cette approche, une méthode de régularisation, variante du Group Lasso, force à zéro les poids des connexions entre le neurone de sortie et les régulateurs les moins actifs. Les poids des connexions partant des gènes d'entrée impliqués dans ces mécanismes de régulation les moins actifs seront également nullifiés. La méthode de régularisation permet également de détecter les groupes de covariables, c'est-à-dire des gènes corégulés qui ont tendance à avoir des expressions corrélées et de faire en sorte que leur implication soit similaire. D'autres travaux se sont basés sur les réseaux de voies métaboliques. Par exemple, dans PasNet (*Pathway-associated sparse deep neural network*) [Hao+18] et GPD (*Gene–Pathway–Disease*) [Gau+20], la première couche cachée du réseau représente les voies métaboliques de l'ontologie Reactome sélectionnées selon leur connectivité avec les gènes d'entrée. Dans PasNet, les poids des connexions de la partie cachée du réseau de neurones ne dépassant pas un certain seuil sont mis à zéro, alors que, dans GPD, aucune régularisation n'a été adoptée. Les auteurs avaient montré dans leurs résultats qu'une régularisation de type L1, L2, ou *dropout* n'avait pas d'impact sur la performance du modèle. Dans P-Net (*Pathway-aware multi-layered hierarchical Network*) [Elm+21] et BioVNN (*Biological Visible Neural Network*) [LL21], les couches du réseau représentent la hiérarchie des voies métaboliques issues de l'ontologie Reactome où les voies métaboliques du sommet de la hiérarchie correspondent aux couches les plus profondes pour s'aligner sur le fonctionnement du réseau de neurones où les couches les plus profondes extraient des caractéristiques de plus haut niveau et souvent plus abstraites. Contrairement aux travaux précédents appliqués sur une seule source de données omiques (*single-omic* en anglais), ces deux modèles ont été appliqués sur plusieurs sources (*multi-omics* en anglais). La différence entre ces deux travaux repose d'une part sur la manière d'intégrer la multimodalité des données et d'autre part sur le nombre de niveaux intégrés de la base de connaissance. Sur le premier point, dans BioVNN, seules deux modalités sont utilisées (données d'expression, le statut de délétion des gènes) et elles sont simplement concaténées. Il est possible qu'un gène y soit représenté deux fois. Dans P-Net, différentes données moléculaires en entrée sont intégrées : les données d'expression de gènes, de méthylation, de mutation et le nombre de copies. La couche d'entrée représentant usuellement les gènes correspond ici à la première couche cachée du réseau afin de combiner les différentes sources. Sur le second point, tous les niveaux de la hiérarchie sont représentés dans BioVNN alors que dans P-Net, seuls les cinq premiers niveaux de la hiérarchie représentant des voies métaboliques plus générales sont inclus. Le choix n'est pas clairement justifié et pourrait s'expliquer par un problème de passage à l'échelle. Dans ces deux réseaux, des connexions résiduelles existent entre chaque couche intermédiaire du réseau et la couche de sortie pour permettre de prédire plus tôt. Aucune étude d'impact sur la présence ou l'absence de ces connexions résiduelles n'a été menée. Enfin, deux travaux que sont GONN *(Gene Ontology Neural Network)* [PWS19] et ParsVNN *(Parsimony Visible Neural Network)* [Hua+21] ont intégré l'ontologie de gènes GO, mais de manière différente. Dans GONN, seule une couche cachée représente un niveau des sous-ontologies GO-MF et GO-BP. Au contraire, ParsVNN intègre une fraction de la hiérarchie où une fonction GO est représentée par six neurones qui correspondent au nombre optimal défini dans leurs précédents travaux [Kue+20]. Pour eux, l'information contenue par une fonction GO ne peut être capturée que par un seul neurone. Selon une technique de compression de la taille du réseau de neurones, les connexions de poids peu élevés sont élaguées et, par conséquent, les neurones les moins informatifs. Tout en minimisant la perte de performance, un sous système de GO est ainsi construit automatiquement et de manière parcimonieuse pour chaque type de cancer considéré. Un résumé comparatif de ces méthodes se trouve dans la Table 3.2.

Ces approches peuvent être discutées. En premier lieu, seuls les gènes reliés à la connaissance sont en général pris en compte, excepté dans GONN où les gènes non annotés sont connectés à des neurones additionnels concaténés à la couche de connaissances. Ces gènes non définis biologiquement pourraient contenir de l'information utile pour le problème à résoudre. Les inclure

permettrait de faire de la découverte biologique. En second lieu, dans les travaux incluant différents niveaux d'une hiérarchie de connaissances, les neurones de la couche d'entrée sont en général seulement connectées à la première couche cachée [Elm+21 ; LL21]. De ce fait, les relations des neurones d'entrée avec les concepts biologiques associés aux neurones des couches profondes sont exclues, de même que les relations non adjacentes entre les couches profondes. Une part de la connaissance est ainsi omise. Seul dans ParsVNN, les gènes d'entrée ont été connectés aux neurones plus profonds par des connexions résiduelles. Néanmoins, seules 2086 fonctions GO sont incluses sur plus de 20 000. Le type d'architecture du modèle restreint l'intégration de certaines connaissances. Par définition, un MLP ne peut représenter qu'un DAG qui ne correspond pas à la structure de certains graphes tels que KEGG et STRING. De plus, les annotations sur les arcs ne peuvent pas être représentées comme le sens de la régulation (sous-régulation, sur-régulation) dans [Kan+17]. En troisième lieu, une validation de l'association neurone-concept biologique n'est généralement pas effectuée. En omettant les relations non adjacentes, il est fort probable que les neurones des couches les plus profondes ne représentent pas les concepts biologiques sous-jacents. Dans GONN et PasNet, une couche cachée non liée à des concepts biologiques a été ajoutée pour extraire des relations non-linéaires entre les concepts biologiques de la couche précédente. Cela a pour effet de rendre le modèle moins explicable. Enfin, la plupart de ces méthodes se focalise d'abord sur les performances plus que sur l'interprétation. L'interprétation menée consiste généralement à établir un classement des neurones et de leurs concepts biologiques associés à partir de l'analyse des poids des connexions (GONN, GRRANN, PasNet, BioVNN). Cela permet de déterminer quels sont les concepts biologiques les plus décisifs. Cette analyse est limitée, car cela va dépendre fortement de la valeur d'activation des neurones et de la prédiction. Certains travaux (P-Net et GPD) ont donc eu recours aux méthodes post hoc pour interpréter de manière plus rigoureuse les résultats de leur modèle. Un score est ainsi attribué à chaque neurone du réseau pour permettre de repérer les concepts biologiques importants. Dans les deux cas, l'interprétation se fait majoritairement au niveau global ou semi-local (à l'échelle d'un groupe d'individus) et non pas à l'échelle d'un individu.

Méthode	Source	Dataset	Problème	KG	#Couches	Régularisation
GRRANN [Kan+17]	single-omic (GE)	GEO	réponse au traitement	STRING	1	Group Lasso
PasNet [Hao+18]	single-omic (GE)	TCGA	survie	Reactome	1(+1)	dropout, élagage, L2
GPD [Gau+20]	single-omic (GE one-hot)	[Cog+08]	diagnostic	Reactome	1	-
P-Net [Elm+21]	multiomics (GE, A, D, M)	[Elm21]	diagnostic	Reactome	6	RC
BioVNN [LL21]	multiomics (GE, D)	CCLE RNA, CRISPR	dépendance des gènes	Reactome	13	RC, batch norm, dropout, L2
GONN [PWS19]	single-omic (GE)	SRP041736 [Pol+14]	type cellulaire	GO	1(+1)	L2
ParsVNN [Hua+21]	single-omic (M)	CRTP,GDC	survie/réponse au traitement	GO	NaN	élagage

TABLE 3.2 – *État de l'art des méthodes FFNN contraintes.* *Les colonnes "Source" et "Dataset" indiquent respectivement le nombre de sources de données moléculaires (single ou multi) et le jeu public utilisés. "M" signifie données de mutation, "GE" d'expression, "D" de délétion et "A" d'amplification. La colonne "Problème" définit la tâche de prédiction de phénotype considérée. Dans certains cas, il s'agit d'un problème de biologie comme la prédiction de la dépendance des gènes ou du type cellulaire. La colonne "#Couches" indique le nombre de couches cachées intégrant la connaissance qui est précisée par la colonne "KG". (+1) rapporte qu'une couche cachée non basée sur la connaissance est ajoutée par-dessus alors que "NaN" signifie que le nombre de couches n'a pas été renseigné. La colonne "Régularisation" informe des techniques de régulation employées. "RC" fait référence aux connexions résiduelles.*

ii) Méthodes basées sur les graphes La seconde alternative consiste à proposer des méthodes pour traiter directement des données prenant la forme d'un graphe (orienté ou non orienté). Cela pourrait résoudre les limites évoquées par le second et le troisième point précédent. Cela ne dépend que de la capacité des méthodes à traiter ce type de données non-euclidiennes. Ainsi, un récent type de réseaux de neurones est apparu pour traiter ce type de données, qu'on nomme les réseaux de neurones basés sur les graphes. L'idée principale de ces nouveaux modèles est de propager et d'agréger l'information contenue par les nœuds (représentés par des neurones) et les arêtes d'un graphe vers leur voisinage en vue de résoudre une tâche de prédiction. Ce type de modèle peut être utilisé pour des tâches de classification de nœud, d'arc et de graphe. Dans le cas d'une tâche de classification de graphe, le graphe est généralement réduit au fur et à mesure. Les GNNs ont déjà été utilisés dans des applications biologiques pour prédire l'étiquette de molécule, d'atome ou de liaison [Jin+21 ; Zha+21a]. Peu de travaux ont été publiés pour la prédiction du phénotype sur les données moléculaires. La prédiction de phénotypes est un problème de classification de graphes où un patient est représenté comme un graphe dont les nœuds contiennent l'information sur le profil moléculaire du patient. La plupart des travaux utilisent des graphes d'interactions génétiques non dirigés tels que les réseaux IPP [RSK18 ; Ram+20 ; Che+21] ou les graphes de coexpression [Ram+20 ; Xin+21] pour définir leur couche d'entrée. Comme illustré par la Fig. 3.7, les données sont ensuite propagées vers des couches convolutionnelles et de réduction de graphe, et enfin des couches entièrement connectées et une de sortie. Ces approches visent principalement à maximiser la précision et à surpasser l'état de l'art. Elles se basent sur des GNNs qui ne sont pas facilement explicables [KW17 ; DBV16].

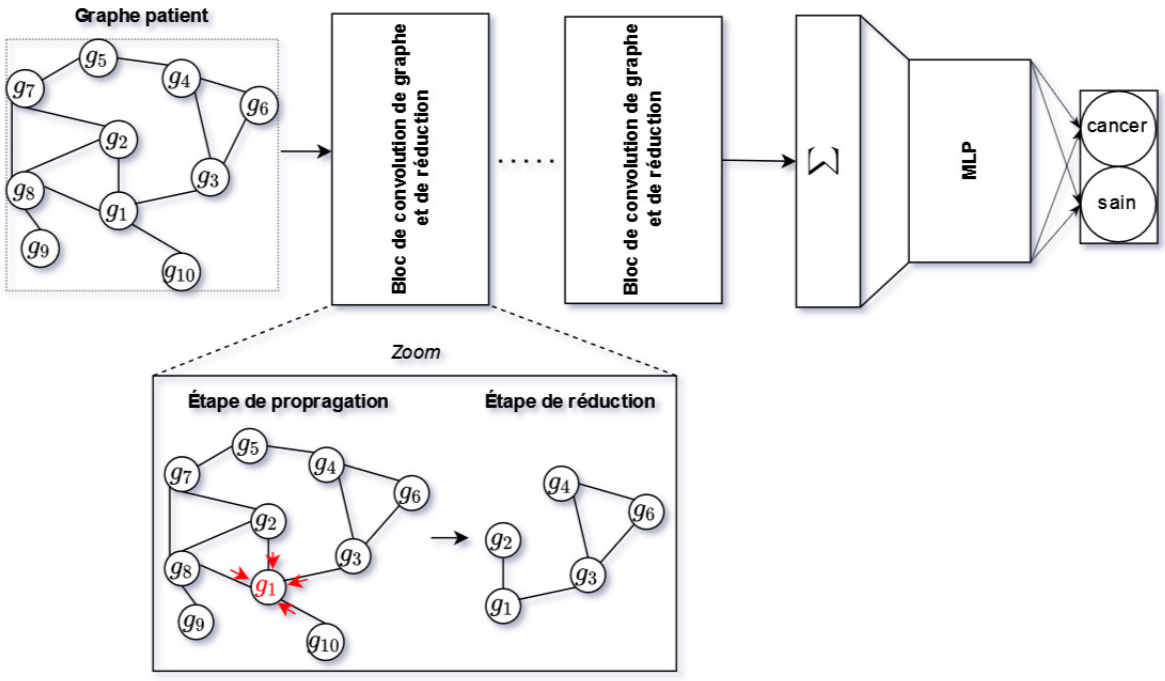


FIGURE 3.7 – ***Illustration du fonctionnement d'un GNN appliqué à un réseau IPP pour une tâche de classification de graphe (diagnostic médical).*** *Le graphe d'entrée IPP peut être traité par plusieurs blocs consécutifs de convolution et de réduction de graphe, similairement à ce qui est fait dans un réseau convolutif. L'étape de convolution consiste à mettre à jour l'information portée par chaque nœud à partir de l'information provenant de leur voisinage (par exemple, composé des nœuds $\{g_2, g_3, g_8, g_{10}\}$ pour le gène g1). Comme pour l'opération de concaténation dans un CNN, la dernière étape avant d'appliquer des couches denses consiste à résumer l'information du graphe entier (représentée ici par le symbole $\sum$). Cela peut être fait de différentes manières, par exemple soit en prenant la somme, la moyenne ou encore le maximum de l'information portée par tous les nœuds du graphe.*

iii) Utilisation de plusieurs sources de connaissances La biologie étant complexe, se contenter d'une seule source de connaissances ne suffit pas forcément. Quelques méthodes tentent ainsi d'intégrer différents niveaux de connaissances qui ont l'avantage d'être interconnectés. Snow et al. ont proposé un réseau de neurones à propagation avant, intitulé BDKANN (*Biological Domain Knowledge-based Artificial Neural Network*), utilisant consécutivement deux sources différentes de connaissances. Dans ce réseau, la première couche cachée représente des complexes protéiques, la seconde couche des voies métaboliques [Sno+21]. Il est intéressant de noter que dans le même article, les auteurs ont également développé une seconde version de BDKANN, intitulée BDKANN+. Dans cette version, les connexions du réseau de neurones ne correspondant pas à des relations dans la base de connaissances ne sont pas masquées, mais pénalisées par une régularisation L1 pour permettre aux connexions véhiculant de l'information intéressante pour le problème de s'exprimer (illustré par la Fig. 3.8). Concernant les GNNs, quelques approches existent également fonctionnant sur des graphes dits hétérogènes. À l'inverse des graphes homogènes contenant un seul type de nœud et d'arête sur lesquels les GNNs sont généralement appliqués, les graphes hétérogènes incluent au moins deux types d'arc ou deux types de nœud différents. Ces graphes peuvent de la sorte représenter l'inter-connectivité entre différentes bases de connaissances. Par exemple, [Wan+20] ont intégré trois types de nœuds différents : les composés chimiques, les gènes et les voies métaboliques (voir Fig. 3.9). On distinguera deux catégories de connexions : les connexions intra au sein d'un même graphe de connaissances et les connexions transversales ou inter entre deux graphes de connaissances. Chaque catégorie contient jusqu'à trois types de connexions différentes.

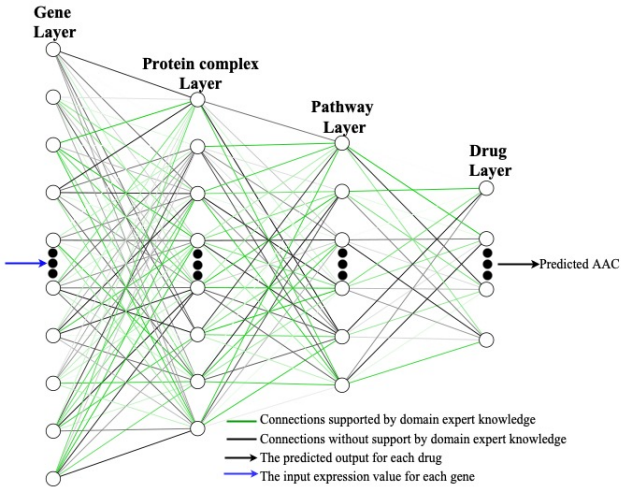


Figure 3.8 – ***Approche de Snow et al. [Sno+21] intégrant deux sources de connaissances différentes (sous licence CC).***

Peu de méthodes sont *self-explaining* qu'elles soient à propagation avant ou basées sur les graphes. Une interprétation a posteriori reste en général nécessaire pour pouvoir identifier le sous-ensemble de neurones et leurs concepts biologiques associés ainsi que le sous-ensemble de variables d'entrée les plus importants pour la prédiction. Dans le cas des modèles à propagation avant, [Gau+20 ; Elm+21 ; Sno+21] ont eu recours aux méthodes d'attribution (analyse de sensibilité, méthode de décomposition, méthode de perturbation) pour identifier les gènes et les concepts biologiques de plus grande importance. Certains travaux sur les GNNs essaient également d'inspecter leurs modèles dans une analyse a posteriori pour les rendre explicables. Cela passe généralement par une étape d'adaptation de méthodes d'interprétation a posteriori [Yua+20] comme les méthodes de décomposition pour qu'elles puissent fonctionner avec ce nouveau type d'architecture [Ram+20 ; Xin+21 ; Che+21].

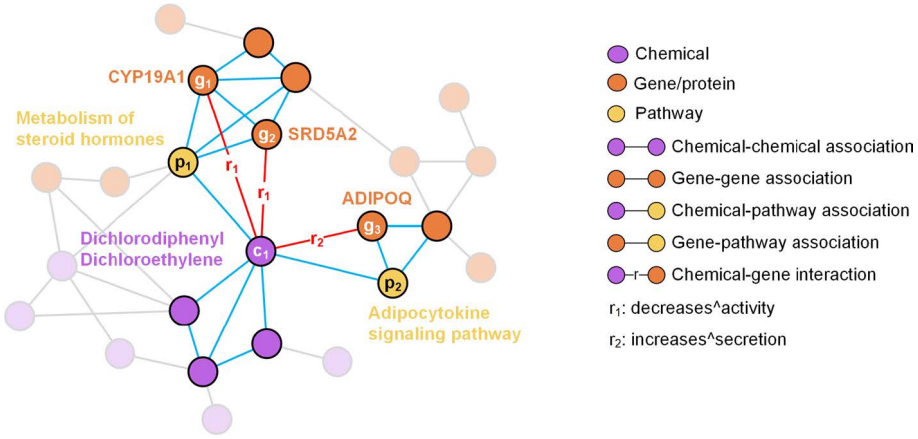


FIGURE 3.9 – ***Exemple de graphe hétérogène utilisé dans* WANG *et al. [Wan+20] (sous licence CC).***

Chap.

4 Deep GONet

Contenu du chapitre

Le premier travail de ma thèse a consisté à la mise au point d'un nouveau réseau de neurones, interprétable par construction, à propagation avant (FFNN) et contraint par la connaissance. Les principaux objectifs étaient premièrement d'intégrer plusieurs niveaux d'une base de connaissances, là où la plupart des méthodes de l'état de l'art n'en intégraient qu'une seule, deuxièmement, d'inclure l'ensemble des gènes annotés et non annotés pour rendre possible de nouvelles découvertes biologiques alors qu'en général seule une poignée de gènes annotés est considérée, et enfin d'offrir une étude de cas détaillée pour montrer comment utiliser le modèle en pratique, souvent manquante dans les précédents travaux. Ce travail a fait l'objet d'une publication [1] dans le journal BMC Bioinformatics [Bou+21c], ainsi qu'une présentation orale lors de la conférence internationale de bio-informatique APBC2021 [Bou+21a] et de la conférence nationale d'apprentissage automatique CAP2021 [Bou+21b].

Dans un premier temps, la méthode sera présentée. Puis, nous analyserons les résultats d'interprétation obtenus ainsi que les performances acquises sur des jeux de données réelles d'expression de gènes. Enfin, nous discuterons de ce premier travail et des perspectives.

4.1 Méthode

Ce modèle, intitulé Deep GONet, est tout d'abord basé sur un MLP simulant une partie de la structure de l'ontologie GO, détaillé en sous-section 4.1.1. Les contraintes sont introduites dans le réseau à l'aide d'un terme de régularisation adapté décrit en sous-section 4.1.2. Enfin, un score d'interprétation sera utilisé pour compléter l'interprétation qui sera introduite en sous-section 4.1.3.

1. Code disponible sur https://forge.ibisc.univ-evry.fr/vbourgeais/DeepGONet.

4.1.1 Description de l'architecture

Notre modèle reçoit dans la couche d'entrée le profil d'expression génétique d'un patient et renvoie dans la couche de sortie la prédiction d'un phénotype de ce patient. L'architecture des couches cachées représente une sous-structure de GO, qui est un graphe acyclique dirigé (DAG). Chaque nœud est un terme GO représentant une fonction biologique par exemple. Deux termes GO u et v sont liés si leurs fonctions sont apparentées. La relation se matérialise alors selon l'arc : $u \rightarrow v$ où v est le terme parent de u. La majorité de ces relations sont de type $''is\_a''$. Par ailleurs, les termes GO sont connectés en respectant une orientation hiérarchique ascendante. Un terme GO est attribué à un niveau dédié en fonction du chemin le plus long vers la racine (le terme "GO :0008150" représenté au sommet de la Fig. 4.1). L'ontologie possède également un ensemble de feuilles, c'est-à-dire que ces nœuds n'ont pas d'arcs entrants, comme c'est le cas du terme "GO :0014810" situé au 19e niveau dans la Fig. 4.1. Les termes GO des niveaux inférieurs correspondent à des fonctions plus spécifiques, telle que la régulation positive du squelette portée par le terme "GO :0014810", tandis que les nœuds de niveau supérieur sont des fonctions plus générales. Les termes GO sont aussi liés aux gènes via des annotations GO. Selon le principe de transitivité, un terme GO hérite de l'ensemble des gènes de ses termes GO fils.

L'architecture de notre réseau de neurones représente une sous-ontologie de GO, c'est-à-dire que chaque couche cachée l représente un niveau GO h, chaque neurone un terme GO et chaque variable d'entrée un gène. La sélection des niveaux constitue un des hyperparamètres du modèle à déterminer. Dans l'exemple de la Fig. 4.1, les niveaux 7 à 2 ont été choisis, cette sélection étant délimitée par l'encadré en vert.

Notre modèle est basé sur un MLP totalement connecté, précédemment introduit par la sous-section 2.1.1. Celui-ci consiste d'une couche d'entrée, de couches cachées et d'une couche de sortie pour la prédiction de phénotypes. La couche d'entrée est composée de gènes ou de fragments de gènes. Chaque neurone est connecté à tous les neurones de la couche précédente et à tous les neurones de la couche suivante. Chaque couche cachée correspond à un niveau GO et ses neurones correspondent à tous les termes GO du niveau cible. L'incorporation des connaissances doit respecter l'objectif du réseau de neurones, qui est de construire une représentation abstraite des données grâce à son architecture hiérarchique. La première couche cachée d'un réseau de neurones extrait les caractéristiques de bas niveau de la couche d'entrée, elle correspond au niveau sélectionné le plus bas de GO contenant des termes plus spécifiques. Dans les dernières couches cachées, les caractéristiques de haut niveau représentent les termes les plus généraux des niveaux GO les plus élevés. Le bas de la Fig. 4.1 illustre comment les niveaux 7 à 2 ont été intégrés dans l'architecture de Deep GONet. Les détails mathématiques sont les mêmes qu'énoncés en introduction dans la sous-section 2.1.1.

Dans notre architecture densément connectée, nous identifions deux types de connexions :

— des connexions correspondant à des relations ou annotations dans GO (colorées en rouge dans la Fig. 4.1), appelées connexions GO ;

— des connexions supplémentaires ne représentant aucunes relations ou annotations dans GO (marquées par des flèches en pointillé), appelées connexions noGO.

Un gène dans la couche d'entrée est connecté aux neurones de la première couche cachée par une connexion GO s'il est associé au terme GO correspondant dans le niveau le plus bas choisi (c.-à-d. le niveau 7 dans la Fig. 4.1), ou par une connexion noGO sinon. Notons que les neurones des couches cachées suivantes (c.-à-d. 2 à 6) ne sont pas directement connectés aux gènes. Ces neurones peuvent être indirectement connectés à leurs gènes par la propagation de l'expression génique à travers les connexions GO des couches précédentes. Si nous voulons représenter exactement la sous-ontologie choisie, nous pouvons couper toutes les connexions noGO et ne garder que les connexions GO dans notre architecture. Cependant, l'ontologie ne représente que les connaissances actuelles que nous avons en biologie. Les ontologies sont mises à jour continuellement du fait de nouvelles découvertes scientifiques. Certains liens peuvent être

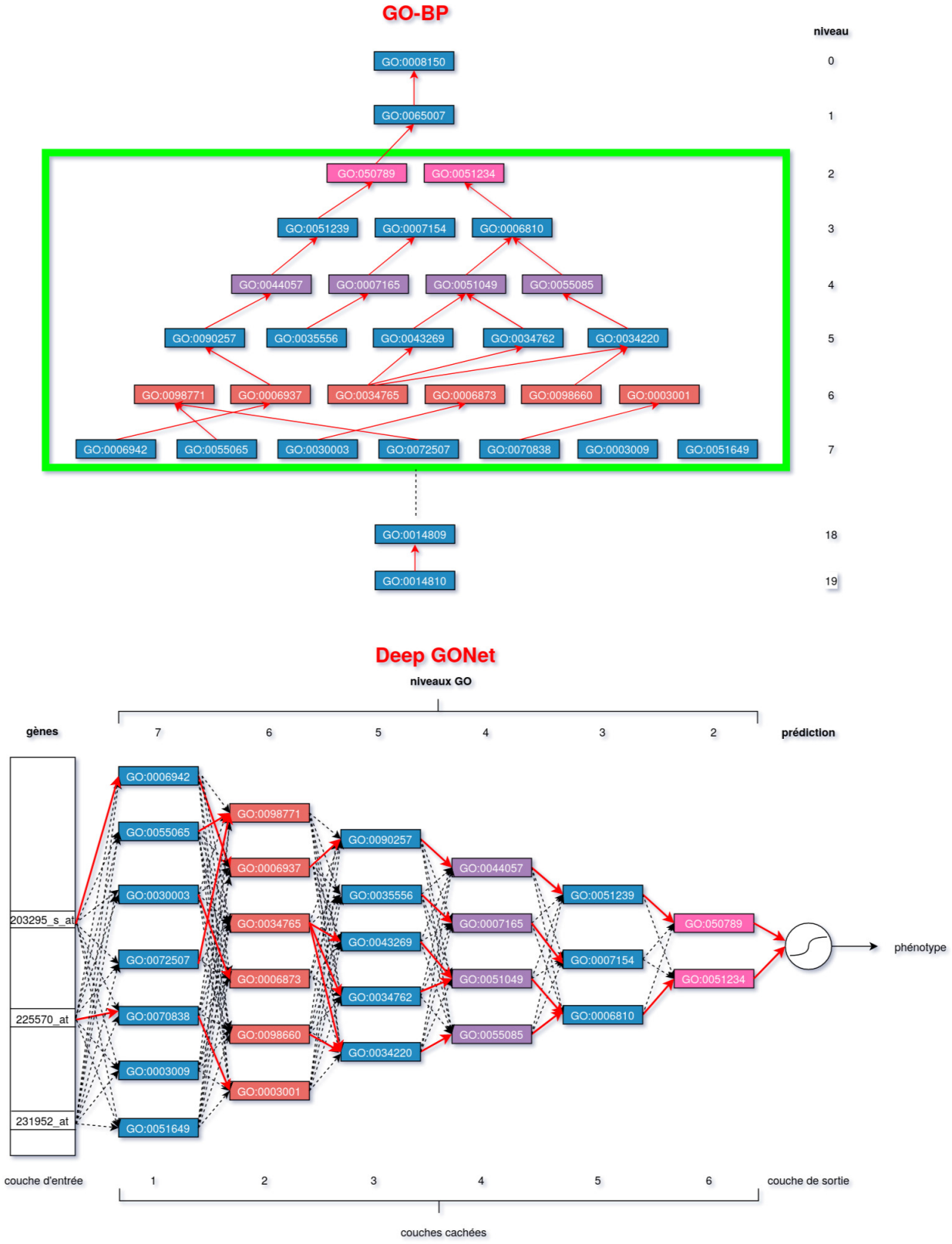


FIGURE 4.1 – ***Aperçu en haut d'une sous-ontologie tronquée de GO (GO-BP) et en bas de l'architecture de Deep GONet.*** *Les niveaux de GO implémentés dans Deep GONet se situent dans l'encadré vert. Les flèches rouges et noires en pointillé représentent respectivement les connexions GO et noGO.*

manquants ou erronés, et de nombreux gènes ne sont pas associés à un terme GO (comme le fragment "231952_at" dans la Fig. 1). Cela signifie que si nous utilisons uniquement les connexions GO, ces gènes ne seront pas connectés au réseau de neurones, mais ils peuvent contenir de l'information intéressante pour le problème traité. Cette situation pourrait avoir un impact négatif sur la capacité de prédiction du réseau de neurones. Pour faire face aux erreurs et à l'incomplétude des connaissances représentées dans GO, nous avons choisi de conserver toutes les connexions dans notre architecture, aussi bien les connexions GO que les connexions noGO. Toutefois, les connexions noGO sont pénalisées afin de favoriser l'utilisation des connexions GO pour calculer les prédictions.

4.1.2 Apprentissage et régularisation du réseau

Le modèle est contraint par un terme de régularisation adapté, appelé L_{GO}, pour favoriser les connexions GO et pénaliser les connexions noGO. Ce terme de régularisation est défini comme suit :

$$L_{GO} = \sum_{l=1}^{L} \|W^{(l)} \otimes (1 - M^{(l)})\|_2^2 \tag{4.1}$$

où $W^{(l)}$ est la matrice de poids de la couche l et $\otimes$ le produit d'Hadamard. $M^{(l)}$ est la matrice d'adjacence qui encode les relations entre les termes GO représentés dans les couches $(l-1)$ et l (de niveaux correspondants $(h+1)$ et h). Plus précisément, si un terme i de niveau correspondant h dans GO est parent du terme j de niveau $(h+1)$, alors $m_{j,i}^{(l)} = 1$, sinon $m_{j,i}^{(l)} = 0$. Pour la couche de sortie, $M^{(L)}$ est une matrice de uns. La fonction de coût est alors composée de l'entropie croisée et de ce régularisateur :

$$L = \sum_{i=1}^{N} \sum_{c=1}^{C} (-y_{i,c} \log \hat{y}_{i,c}) + \alpha L_{GO} \tag{4.2}$$

où N et C correspondent respectivement au nombre d'exemples dans le jeu d'apprentissage et au nombre de classes. $y_{i,c}$ est l'indicateur de la classe positive, c.-à-d. $y_{i,c} = 1$ quand le i-ème exemple appartient à la classe c, ou 0 sinon. Dans notre cas, chaque exemple appartient seulement à une seule classe. $\hat{y}_{i,c}$ est la probabilité calculée par Deep GONet que le i-ème exemple appartienne à la classe c. Pendant la phase d'inférence, nous sélectionnons la classe avec la probabilité la plus élevée pour déterminer la prédiction finale. Enfin, α est un hyperparamètre qui pondère le terme de régularisation. Avec une valeur proche de 0, le terme de régularisation disparaît, notre modèle devient un MLP classique sans capacité d'interprétation. Avec une valeur élevée, l'algorithme d'apprentissage se concentre sur le terme de régularisation et ignore l'entropie croisée. Le réseau de neurones résultant représente parfaitement les connexions GO en ignorant les connexions noGO, mais il a une faible capacité de prédiction. α s'avère être un compromis entre la minimisation de l'entropie croisée (performance) et du régularisateur L_{GO} (interprétation). Il est donc important de bien ajuster la valeur prise par cet hyperparamètre.

4.1.3 Interprétation a posteriori

Pour identifier les neurones utilisés pour calculer les prédictions et leurs concepts biologiques associés, nous aurons recours à la méthode de décomposition LRP [Bac+15 ; MSM18], illustrée par la Fig. 4.2. L'objectif de LRP est de rétropropager le signal de sortie d'un exemple de la couche cachée supérieure vers la couche d'entrée. Le score de pertinence attribué à un neurone i d'une couche l est donné par la formule suivante, correspondante à la variante LRP-ϵ :

$$R_i^{(l)} = \sum_{j=0}^{N_{l+1}} R_{i \leftarrow j}^{(l+1)} = \sum_{j=0}^{N_{l+1}} \frac{a_i^{(l)} w_{i,j}}{\sum_k a_k^{(l)} w_{k,j} + \epsilon} R_j^{(l+1)} \tag{4.3}$$

où ϵ est un facteur de stabilisation pour éviter les divisions par zéro, $R_{i\leftarrow j}^{(l+1)}$ représente la relevance de la connexion du neurone j de la couche $(l+1)$ au neurone i de la couche inférieure l, $R_j^{(l+1)}$ la relevance du neurone j de la couche supérieure $(l+1)$, et $R_i^{(L)} = z_i^{(L)}$. En fixant ϵ à 10^{-7} dans nos expériences, la propagation de la relevance est conservée au travers du réseau, c'est-à-dire que la somme des scores des variables d'entrée est égale à la valeur de pré-activation de la sortie telle que $\sum_i R_i^0 = z_c^{(L)}$.

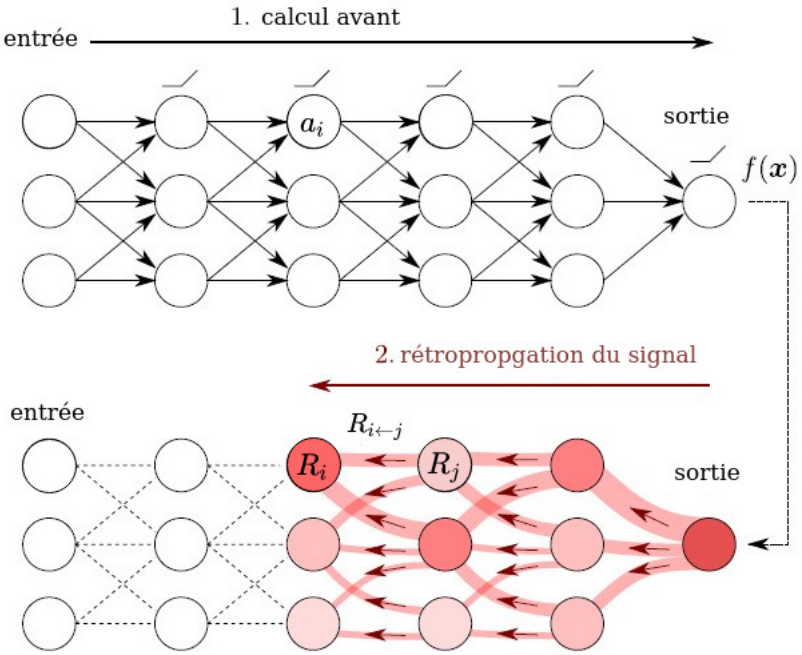


FIGURE 4.2 – ***Procédure LRP schématisée et traduite depuis [MSM18] (sous licence CC).***

Méthode	Score R_i^0
SA [SVZ14]	$\left\lvert \frac{\partial a_c(x)}{\partial x_i} \right\rvert$
GI [SVZ14]	$x_i \times \frac{\partial a_c(x)}{\partial x_i}$
LRP-ϵ [Mon+19]	$\sum_j \frac{x_i w_{ij}}{\sum_k x_k w_{kj} + \epsilon} R_j^0$, quand $\epsilon = 0 \equiv$ LRP-z (LRP original)
LRP-$\alpha\beta$ [Mon+19]	$\sum_j \left(\alpha \frac{x_i w_{ij}^+}{\sum_k x_k w_{kj}^+} - \beta \frac{x_i w_{ij}^-}{\sum_k x_k w_{kj}^-}\right) R_j^0$ sujet à $\alpha - \beta = 1$ avec $\beta \geq 0$, quand $\alpha = 1, \beta = 0 \equiv$ LRP-γ
IG [STY17]	$(x_i - \tilde{x}_i) \cdot \int_0^1 [\nabla f(\tilde{x} + t \cdot (x - \tilde{x}))]_i dt$

TABLE 4.1 – ***Description du score de pertinence R_i^0 pour une variable d'entrée i selon différentes méthodes d'attribution.*** *f désigne ici un réseau de neurones.*

Ce choix a été motivé pour différentes raisons. Tout d'abord, LRP est une méthode d'interprétation facile à mettre en place et peu coûteuse. Une seule passe arrière est nécessaire pour calculer ces scores. Des études ont également montré que les résultats produits avec LRP sont satisfaisants et restent alignés sur l'interprétation humaine [MSM18 ; Sam+19]. Ensuite, la version ϵ choisie fonctionne bien sur des réseaux de neurones totalement connectés [Koh+20b]. Une

précédente étude [Han+20] menée sur les données d'expression de gènes a confirmé que les résultats produits par cette version sont corrélés à ceux obtenus par d'autres méthodes d'attribution (SA, GI, IG, LRP-$\alpha\beta$) (voir en annexe 7.1). Les formules de ces méthodes sont listées dans la Table 4.1. L'avantage de LRP sur les méthodes SA et GI est qu'elle s'avère moins sensible aux bruits [MSM18]. La méthode IG, quant à elle, nécessite de définir une référence dénommée $\tilde{x}$, difficile à estimer dans le cas de données d'expression de gènes. Dans l'étude [Han+20], les neurones d'entrée avaient été mis à zéro. Par ailleurs, cette méthode de décomposition est plus coûteuse que les autres méthodes puisqu'elle nécessite 50 à 200 passes arrière pour calculer l'intégrale de la formule donnée par la dernière ligne de la Table 4.1, contre une seule passe arrière dans le cas des autres méthodes [Anc+19]. La variante $\alpha\beta$ de LRP balance le signal en fonction du signe de la pondération portée par les connexions sortantes des neurones au moyen des deux hyperparamètres et devient donc plus fastidieuse à paramétrer que LRP-ϵ. Notons qu'en fixant ϵ à 0 (désignée de variante z), LRP devient équivalente à la méthode GI en cas de non-linéarité de type ReLU.

4.2 Résultats

4.2.1 Données utilisées

Comme précédemment introduit, GO regroupe trois sous-ontologies qui ont une structure de DAG : GO-BP, GO-MF et GO-CC. Nous avons choisi de baser l'architecture des couches cachées de Deep GONet sur la sous-ontologie GO-BP. Celle-ci fournit des processus biologiques impliqués par l'activité des gènes, ce qui peut être plus utile pour la prédiction de phénotypes. Cependant, il est tout à fait possible de représenter à la place de GO-BP les sous-ontologies GO-MF ou GO-CC comme elles sont structurées de la même façon sur un DAG. La sous-ontologie GO-BP contient originalement environ 29K termes GO répartis sur une vingtaine de niveaux avec un degré moyen de 2,42, soit une connectivité parcimonieuse de l'ordre de 0,008 %. Le nombre exact de nœuds varie légèrement mensuellement. Nous avons validé notre modèle sur les jeux de données publics microarray et TCGA, décrits dans le chapitre introductif en sous-section 2.2.2 et 2.2.3. Notons que respectivement 33 % et 67,4 % des gènes d'entrée de microarray et TCGA n'ont pas d'annotations GO. Uniquement les termes GO annotés aux gènes des jeux données utilisés ont été finalement retenus.

4.2.2 Analyse de sensibilité

Dans cette première expérience, nous comparons les performances de Deep GONet avec l'état de l'art sur la prédiction de cancer à partir du profil d'expression génétique. La classification binaire est évaluée sur les données microarray avec une fonction sigmoïde en couche de sortie, tandis que la classification multiclasses est effectuée sur les données RNA-Seq avec une fonction softmax.

Le modèle Deep GONet est appris à partir de l'ensemble d'apprentissage en utilisant une procédure d'apprentissage standard. Le nombre de couches et de neurones sont déterminés à partir des niveaux choisis dans GO-BP. En se basant sur la topologie et la connectivité avec les gènes, nous avons fixé l'architecture avec les niveaux 7 à 2 de GO-BP pour les deux jeux de données. Concernant le choix des hyperparamètres d'entraînement, nous avons opté pour les suivants. Les poids et les biais sont initialisés avec l'initialiseur He. Sur les données microarray, des couches de *dropout* avec une probabilité de $0,6$ sont ajoutées après chaque couche cachée afin de réduire le surapprentissage. Les paramètres du réseau sont optimisés en utilisant Adam avec un pas d'apprentissage adaptatif de 0,001. Sur le jeu de données RNA-Seq, nous choisissons la descente de gradient stochastique avec un momentum égal à 0,9 et un pas d'apprentissage

de 0,001. Le nombre d'époques maximum d'apprentissage est fixé à 600. Différentes valeurs de l'hyperparamètre α, contrôlant le terme de régularisation L_{GO}, sont testés dans l'intervalle $[0, 10^1]$. La performance du modèle est estimée à partir de l'ensemble de test en fonction de la valeur d'α afin d'étudier l'impact de cet hyperparamètre sur les performances du modèle. Notre méthode est comparée à des réseaux de neurones entièrement connectés utilisant une régularisation L_1 ou L_2. Ces termes de régularisation appliquent une pénalité sur toutes les connexions, quelque soit leur type (GO ou noGO). La régularisation L_1 correspond à la valeur absolue de la magnitude des poids $L_1 = \sum_{l=1}^{L} |W^{(l)}|$ et la $L2$ au carré de la valeur des poids $L_2 = \sum_{l=1}^{L} \|W^{(l)}\|_2^2$. Ces termes de régularisation sont également contrôlés par un hyperparamètre α. Enfin, un modèle sans aucune régularisation est testé pour comparaison correspondant au point $\alpha = 0$. Tous ces modèles sont fondés sur la même architecture de base décrite par la Fig. 4.1, et sont appris et testés selon la même procédure décrite ci-dessus. Pour réduire la variabilité des résultats provenant de l'initialisation aléatoire des paramètres des modèles, dix entraînements indépendants ont été réalisés pour chaque valeur d'α et chacune des configurations $\{L_1, L_2, L_{GO}\}$. Toutes les expériences ont été exécutées sur un GPU RTX 2080Ti en utilisant l'environnement Tensorflow v1.12. Les performances de chaque modèle sont estimées à partir de l'ensemble de test selon différentes métriques qui peuvent être :

- le taux de bonnes prédictions (*accuracy* en anglais ou taux d'erreur) : rapport entre le nombre de bonnes (ou mauvaises) prédictions (c.-à-d. la classe prédite correspond à la classe réelle) sur la taille de l'échantillon ;
- la précision : rapport entre le nombre de prédictions correctes pour la classe c (ou positive dans le cadre binaire) et le nombre total de bonnes prédictions ;
- le rappel : rapport entre le nombre de prédictions correctes pour la classe c (ou positive dans le cadre binaire) et le nombre total de prédictions pour cette classe ;
- la F-mesure (ou F1-Score) : combinaison pondérée de la précision et du rappel définie de la sorte : $2 \times \frac{précision \times rappel}{précision + rappel}$;
- le coefficient de corrélation de Matthews (MCC) : prise en compte du déséquilibre des classes ;
- l'aire sous la courbe de la fonction d'efficacité du récepteur (en anglais *area under the receiver operating characteristic (ROC) curve*, abrégée AUC) : mesure du lien entre le taux de faux positifs et de vrais positifs pour différentes valeurs de seuil.
- l'aire sous la courbe précision-rappel (en anglais *area under the precision-recall curve*, abrégée AUPRC) : mesure le compromis entre la précision et le rappel pour différentes valeurs de seuil.

Les formules des métriques dans le cadre binaire sont données dans la Table 4.2. Dans le cas d'un problème multiclasses, on réalise généralement une macro-moyenne pour les mesures de précision, rappel et AUC en calculant d'abord l'ensemble de ces métriques sur chaque classe prise individuellement, puis on réalise la moyenne sur les K classes. Pour la MCC, il existe une formulation spécifique aux problèmes multiclasses [Gor04]. Dans l'ensemble, plus les métriques sont proches de 1, meilleure est le modèle. Au contraire à 0,5 et dans le cas de deux classes, le modèle fait au hasard.

Accuracy	Précision	Rappel	MCC
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

TABLE 4.2 – ***Formules des métriques de performances utilisées dans un cadre binaire.*** *Abréviations : vrais positifs (TP), vrais négatifs (TN), faux positifs (FP), faux négatifs (FN).*

Dans ce qui suit, les résultats sur le jeu de données microarray (Fig. 4.3a-c) ont été commentés, mais des résultats similaires sont observables sur le jeu de données TCGA (Fig. 4.3d-f).

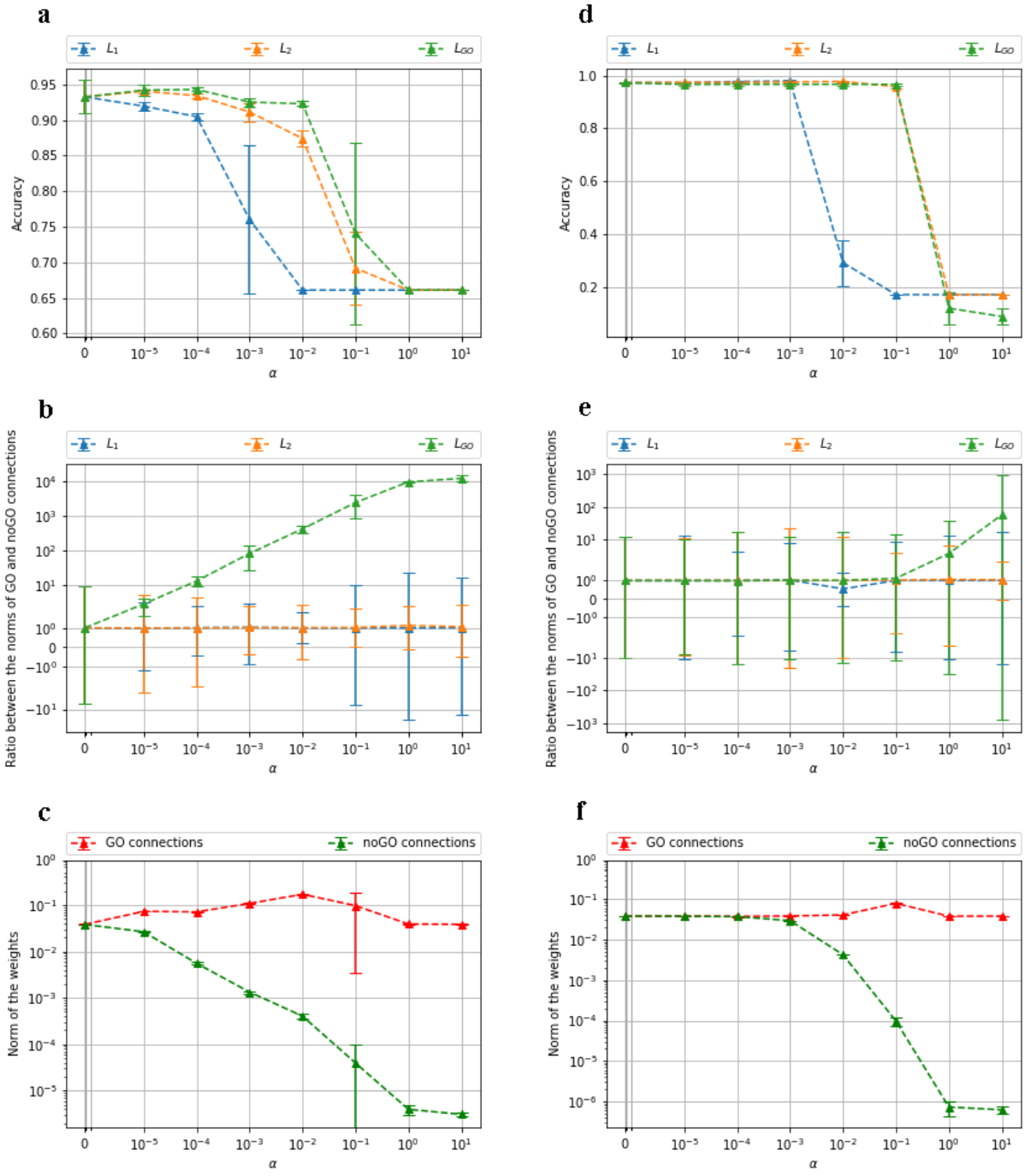


FIGURE 4.3 – ***Résultats sur le jeu de données microarray (colonne de gauche) et TCGA (colonne de droite).*** *a-d Accuracy des modèles selon α. b-e Rapport entre les poids des connexions GO et noGO selon α. c-f Normes de la valeur absolue des poids des connexions GO et noGO des modèles L_{GO} selon α.*

La Fig. 4.3a (resp. Fig. 4.3d) trace la moyenne et l'écart-type de l'accuracy des modèles appris avec une régularisation L_1, L_2 ou L_{GO} selon α. Les trois courbes commencent au même point $\alpha = 0$ correspondant aux modèles sans régularisation. Nous pouvons voir que les modèles sans régularisation et ceux avec une régularisation L_{GO} ou L_2 atteignent la meilleure accuracy de 0,945 à $\alpha = 10^{-5}$. Les modèles construits avec ces dernières régularisations sont plus performants qu'avec une régularisation L_1. Nous avons également testé les méthodes ML suivantes avec le package scikit-learn de Python (les hyperparamètres importants sont indiqués entre parenthèses) : RF (critère de Gini, nombre d'arbres=100), SVM (noyau linéaire, C=1.0), XGboost (nombre d'arbres=100, pas d'apprentissage=0.1), et MLP (trois couches avec respectivement 1000, 500 et 200 neurones). Dix modèles ont été également appris pour chaque méthode. Les Tables 4.3 et 4.4 résument les performances obtenues selon les différentes métriques exposées précédemment. Ces tables montrent qu'avec l'approche Deep GONet, on obtient la même accuracy qu'avec les algorithmes de l'état de l'art, qui rappelons sont opaques. Pour l'ensemble des modèles avec une régularisation L_1, L_2 ou L_{GO}, l'accuracy moyenne diminue pour une valeur élevée d'α. L'accuracy chute à 0,66, correspondant à la proportion de la classe majoritaire, ce qui signifie que les modèles n'ont pas réussi à apprendre et associent tous les exemples à la classe cancer. Dans ce cas, le terme de régularisation prend trop d'importance par rapport à l'entropie croisée. Nous notons quelques points particuliers avec une haute variabilité à $\alpha = 10^{-1}$ (pour les régularisations L_{GO} et L_2) et à $\alpha = 10^{-3}$ (pour L_1). À ces valeurs d'α, certains modèles ne parviennent pas à apprendre lorsqu'ils ont une accuracy de $0,66$, tandis que d'autres réussissent en atteignant une accuracy d'environ $0,9$. C'est pourquoi la moyenne se situe entre ces deux extrêmes.

Dans ce qui suit, nous analyserons le comportement des connexions GO et noGO. La Fig. 4.3b (resp. Fig. 4.3e) montre le rapport entre les normes de valeurs absolues des poids des connexions GO (Eq. (4.4)) et noGO (Eq. (4.5)), définies respectivement de la manière suivante :

$$\frac{1}{L}\frac{1}{\#GO}\sum_{i=1}^{L}|W^{(l)} \otimes (M^{(l)})|, \tag{4.4}$$

$$\frac{1}{L}\frac{1}{\#noGO}\sum_{i=1}^{L}|W^{(l)} \otimes (1 - M^{(l)})|. \tag{4.5}$$

Pour les modèles avec une régularisation L_1 ou L_2, le rapport est bloqué à 1 quelle que soit la valeur d'α. Comme attendu, aucune distinction n'est faite entre les deux types de connexions. Au contraire, le rapport calculé pour les modèles avec une régularisation L_{GO} augmente plus α a une valeur élevée et atteint finalement sa valeur maximale à 10^4. Pour ces modèles, la Fig. 4.3c (resp. Fig. 4.3f) montre la moyenne des normes de valeurs absolues des poids des connexions GO (Eq. (4.4)) et noGO (Eq.(4.5)). On peut noter que la ligne verte de la Fig. 4.3b (resp. Fig. 4.3e) est obtenue par la division de la ligne rouge par la ligne verte de la Fig. 4.3c (resp. Fig. 4.3f). Nous pouvons observer que la norme moyenne des poids des connexions GO reste entre 10^{-2} et 10^{-1}. En revanche, la norme moyenne des poids des connexions noGO diminue avec α, suivant l'allure de l'accuracy. À $\alpha = 0$ et $\alpha = 10^{-5}$, la norme moyenne des poids des connexions noGO est très proche de celle des connexions GO. Le rapport entre ces deux normes, illustré à la Fig. 4.3b, est inférieur à 10^1. De 10^{-4} à 10^1, l'écart entre les deux normes se creuse et la norme des connexions noGO finit par converger presque à zéro. Cela conduit à un rapport de 10^1 à $\alpha = 10^{-4}$ et au rapport le plus élevé de 10^4 à $\alpha = 10^1$. Le terme de régularisation L_{GO} semble ainsi bien pénaliser les connexions noGO avec une valeur élevée d'α. À $\alpha = 10^1$, le modèle peut être considéré comme équivalent à un modèle ne contenant que des connexions GO où toutes les connexions noGO sont mises à zéro, respectant ainsi scrupuleusement la hiérarchie de GO. Cependant, les courbes d'accuracy de la Fig. 4.3a montrent qu'avec une grande valeur d'α, le modèle n'est plus capable d'apprendre. Cela signifie que certaines connexions noGO semblent nécessaires pour obtenir des prédictions correctes. La flexibilité apportée par l'architecture entièrement connectée rend cela possible. Cet avantage sera examiné plus en détail dans les sections suivantes.

En résumé, imposer un nombre de couches et de neurones n'est pas suffisant pour rendre le modèle interprétable. Un terme de régularisation approprié doit être ajouté à la fonction de perte pour guider le modèle sur la base des connaissances biologiques. Si le terme de régularisation n'est pas personnalisé, les connexions GO et noGO seront considérées de manière identique comme avec une régularisation L_1 ou L_2. Il en résulte un modèle opaque sans aucune connaissance intégrée. Notre modèle Deep GONet atteint des performances de prédiction similaires à celles de l'état de l'art, à la fois (i) en pénalisant correctement les connexions noGO, et (ii) en privilégiant suffisamment les connexions GO pour laisser l'information principale passer par ces connexions.

Sur le jeu de données de microarray, les modèles à $\alpha = 10^{-2}$ obtiennent une accuracy moyenne autour de $0,92$ et un rapport moyen de 10^3. Comme ils représentent un bon compromis entre la pénalisation des connexions noGO et l'accuracy, nous analysons en profondeur et interprétons biologiquement l'un des modèles appris à $\alpha = 10^{-2}$ dans la suite de ce chapitre. L'étude se concentrera sur le jeu de données microarray, mais des analyses similaires peuvent être menées sur l'autre jeu de données.

Modèle	Accuracy	Précision	Rappel	F1-Score	MCC	AUC
RF	0,904	0,932	0,921	0,927	0,786	0,895
SVM	0,948	0,964	0,957	0,961	0,885	0,944
XGBoost	0,936	0,954	0,948	0,951	0,857	0,930
MLP	0,951	0,974	0,952	0,963	0,893	0,986
Deep GONet ($\alpha = 10^{-2}$)	0,925	0,943	0,943	0,943	0,832	0,916

Table 4.3 – ***Comparaison des performances des modèles sur le jeu de données microarray.***

Modèle	Accuracy	Précision	Rappel	F1-Score	MCC	AUC
RF	0,968	0,967	0,965	0,966	0,964	0,999
SVM	0,977	0,977	0,975	0,976	0,974	1,000
XGBoost	0,974	0,972	0,972	0,972	0,971	1,000
MLP	0,962	0,961	0,960	0,960	0,958	0,998
Deep GONet ($\alpha = 10^{-1}$)	0,970	0,970	0,967	0,968	0,967	0,998

Table 4.4 – ***Comparaison des performances des modèles sur le jeu de données RNA-Seq.***

4.2.3 Analyse de l'architecture de Deep GONet

La première partie de cette analyse consiste à vérifier que l'architecture du modèle Deep GONet choisi dans la section précédente imite bien de la sous-hiérarchie de GO-BP. Ce modèle a été appris avec $\alpha = 10^{-2}$ et atteint une précision de 0,925 (rapportée dans la Table 4.3) ainsi qu'un rapport entre les connexions GO et noGO autour de 10^3. La Table 4.5 présente en détail l'architecture de Deep GONet. Les deux premières lignes résument les niveaux correspondants de GO-BP et le nombre de neurones de la couche d'entrée à celle de sortie (voir Fig. 4.1). Les deux dernières lignes donnent pour chaque couche le nombre de connexions entrantes (GO et noGO) et le nombre de connexions GO entrantes. Notons que le nombre total de connexions plus le nombre de neurones constituent le nombre de paramètres du modèle (environ 90,105M). Le nombre de connexions diminue à travers les couches, car le nombre de neurones par couche

devient plus petit. Ce tableau montre que la grande majorité des connexions sont des connexions noGO, seulement 0,05 % sont des connexions GO (environ 48K).

La Fig. 4.4 montre pour chaque couche cachée, le classement des connexions entrantes en fonction de la valeur absolue de leur poids. Les connexions GO (resp. noGO) entrantes sont colorées en rouge (resp. vert). Nous remarquons tout d'abord que les matrices de connexions sont très parcimonieuses, peu de connexions ont leur poids significativement différent de 0. Cela signifie que l'expression des gènes n'est pas propagée uniformément à travers le réseau entier et que seule une petite partie du réseau est utile pour la prédiction. Pour l'ensemble des couches cachées, la plupart des connexions GO sont classées avant les connexions noGO. Certaines connexions GO peuvent avoir un poids très élevé (environ 10^2). Les connexions GO entrantes d'un neurone à valeur élevée favorisent l'activation du terme GO correspondant. La valeur des connexions noGO est proche de 0, du fait de l'application de la pénalisation L_{GO}. Certaines connexions GO peuvent être classées en bas du classement. Par exemple, la 43 505$^{\text{e}}$ connexion GO de la première couche est classée à la position 33 041 190. Les connexions GO, qui ne semblent pas être utiles au réseau, obtiennent une valeur très faible (7×10^{-6} pour notre exemple). À l'inverse, du fait de l'application de la régularisation L_{GO} sur les connexions noGO, peu d'entre elles ont un poids plus élevé que les connexions GO comme l'illustre la figure de la deuxième couche cachée. Ces résultats montrent que l'architecture de notre modèle respecte la structure de GO-BP puisque la plupart des poids des connexions noGO sont fixés à 0. Les rares connexions noGO avec un poids élevé sont intéressantes. Elles représentent des liens que le réseau doit construire pour calculer des prédictions précises. Il serait intéressant d'étudier les termes GO ou les gènes qui sont concernés par ces connexions noGO.

Couche	Entrée	1	2	3	4	5	6	Sortie	Total
Niveau GO-BP	–	7	6	5	4	3	2	–	6
#neurones	54 675	1574	1386	951	515	255	90	1	4772
#connexions	–	86M	2,2M	1,3M	490K	131K	23K	90	90,1M
#GO connexions	–	43 504	1709	1585	1010	491	175	–	48K

Table 4.5 – *Détails sur l'architecture de Deep GONet.*

Les prochaines analyses de notre réseau seront basées sur deux ensembles de valeurs : l'activation des neurones et la contribution des neurones. L'activation $a_i^{(l)}$ d'un neurone i de la couche l donne des informations sur la quantité de signal $z_i^{(l)}$ provenant de ce neurone. Cependant, une activation élevée ne signifie pas nécessairement que le neurone contribue fortement à la prédiction. Un neurone fortement activé pour un échantillon donné peut avoir des connexions sortantes avec un poids très faible. Dans ce cas, il contribuera peu à la prédiction. Par conséquent, pour identifier rigoureusement les neurones les plus importants pour les prédictions, nous avons calculé le score de pertinence R_i de chacun des neurones i. Pour chaque patient, nous pouvons obtenir un profil de contribution (resp. d'activation) par couche composé de la contribution (resp. de l'activation) des neurones. Une analyse de l'importance des neurones de chaque couche confirme le fait que seul un petit sous-ensemble de neurones est important pour réaliser une prédiction donnée.

4.2.4 Signification biologique des neurones

Dans cette section, nous vérifions que les neurones de notre réseau représentent effectivement leur terme GO correspondant, c'est-à-dire que l'activation d'un neurone donné représente l'expression de la fonction biologique correspondante. Pour cela, nous utilisons le fait que chaque terme GO dans GO-BP est associé à un ensemble de gènes. Si un neurone donné représente

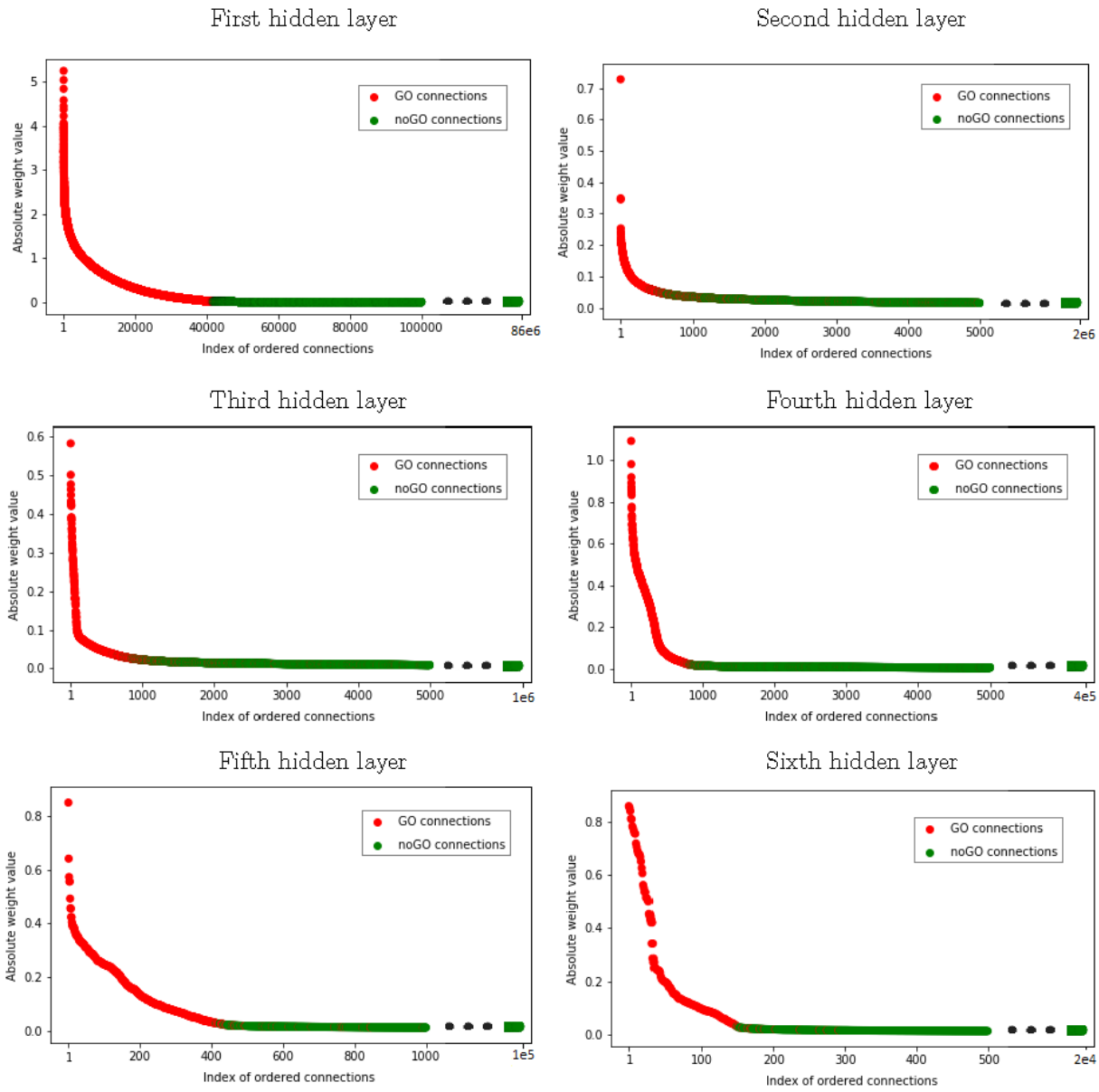


FIGURE 4.4 – ***Classement des connexions entrantes de chaque couche en fonction de la valeur absolue de leurs poids.***

réellement son terme GO correspondant, l'ensemble des gènes associés devrait activer ce neurone plus que tout autre ensemble de gènes. Nous proposons une procédure illustrée dans la Fig. 4.5 pour tester la signification biologique des neurones et évaluer la relation avec leur score de pertinence en utilisant LRP avec le package iNNvestigate [Alb+19]. Nous détaillerons dans ce qui suit uniquement l'analyse de la première couche cachée. Cependant, nous pouvons appliquer des analyses similaires aux autres couches.

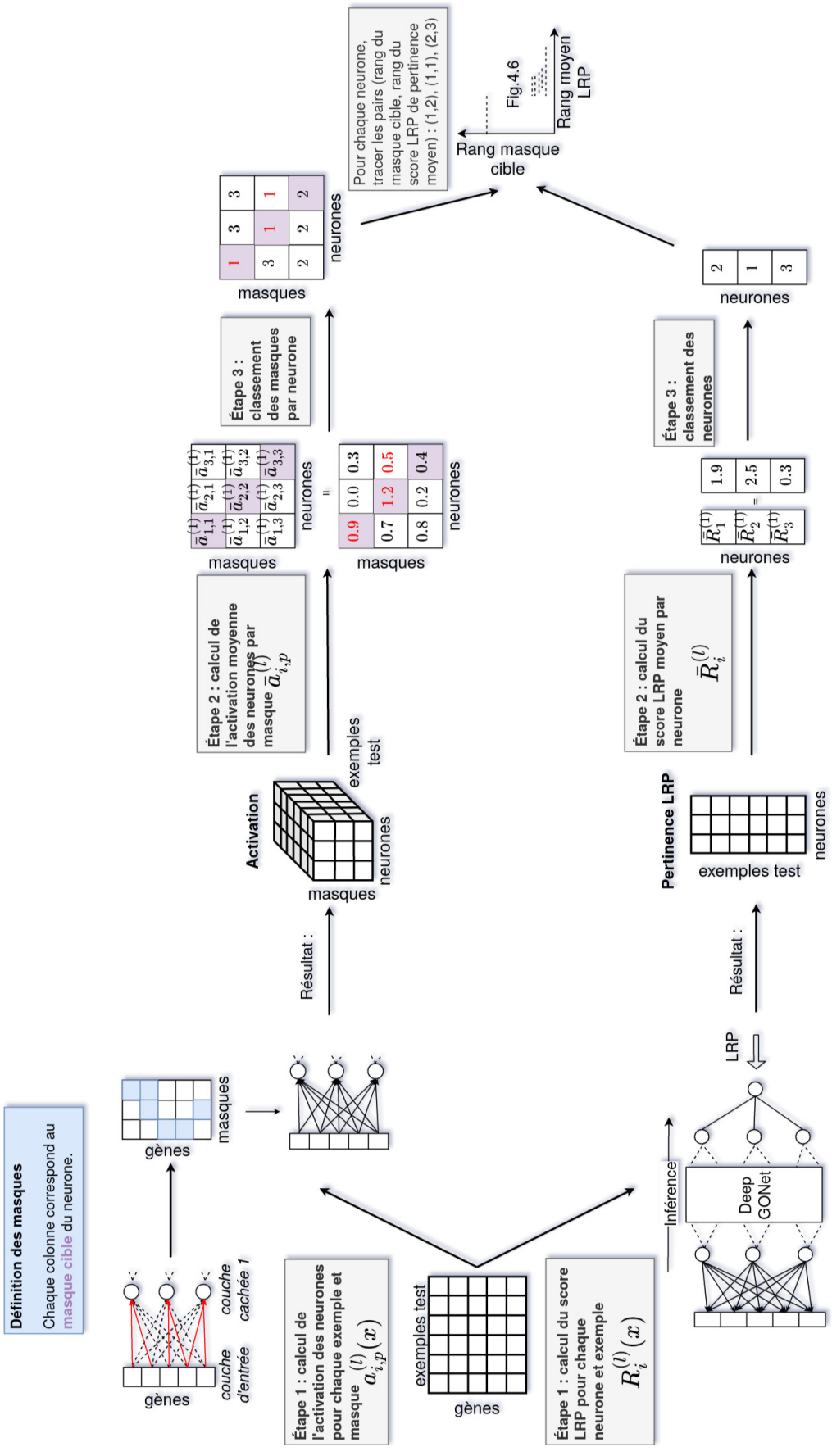


FIGURE 4.5 – *Illustration de la procédure d'évaluation de la signification biologique des neurones de la première couche.* *La partie supérieure montre comment calculer le rang du masque cible de chaque neurone. La partie inférieure montre comment calculer le rang des neurones selon leur score de pertinence. La relation entre ces deux métriques est évaluée par une représentation graphique finale (par exemple, la Fig. 4.6).*

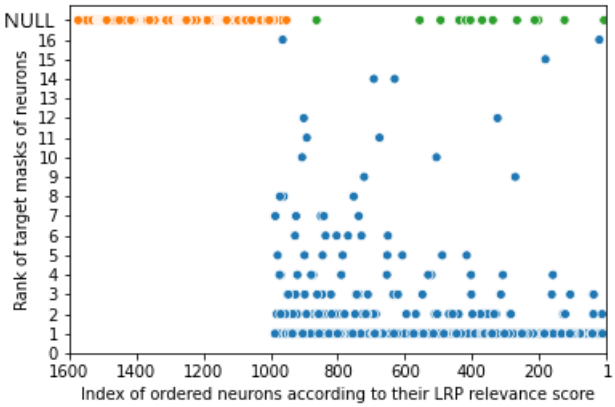


FIGURE 4.6 – ***Classement des neurones de la première couche cachée en fonction du rang de leur masque cible (axe y) et de leur rang LRP (axe x).*** *Les points oranges représentent les neurones jamais activés quel que soit le masque, alors que les points verts représentent ceux n'étant pas activés par leur masque cible, mais par au moins un autre. Les points bleus indiquent au contraire les neurones qui sont activés par leur masque cible.*

La première couche cachée contient 1574 neurones connectés à la couche d'entrée. Chaque terme GO est connecté à un ensemble de gènes (médiane : 8, max : 1357, min : 1). Au regard de ces informations, le masque cible d'un neurone est défini comme suit :

— tous les gènes de la couche d'entrée, qui ne sont pas connectés au terme GO associé, sont mis à 0 ;
— les valeurs des gènes restants sont inchangées.

Au total, nous avons 1574 masques, car aucun des neurones n'a le même masque cible. Pour chaque neurone, tous ses masques sont appliqués sur la couche d'entrée afin d'identifier s'il est plus activé par son masque cible que par les autres masques. Ceci peut être mesuré par le rang du masque cible. La procédure suivante, illustrée en haut de la Fig. 4.5, rapporte comment obtenir le rang du masque cible pour chaque neurone dans une couche l :

— Étape 1 : Pour chaque exemple x de l'ensemble de test, l'activation $a_{i,p}^{(l)}(x)$ de chaque neurone i est calculé sur l'ensemble des masques $m_p^{(l)}$ où $p = 1, \ldots, i, \ldots, N_l$ avec N_l le nombre de neurones dans la couche l. Comme un neurone et son masque cible partagent le même indice, l'activation d'un neurone i pour son masque cible est $a_{i,i}^{(l)}(x)$. Notons qu'il n'y a pas de biais lié à la longueur du masque.
— Étape 2 : On considère ensuite la valeur moyenne de ces activations notée $\bar{a}_{i,p}^{(l)}$.
— Étape 3 : Pour chaque neurone i, les valeurs moyennes sur l'ensemble N_l des masques $\bar{a}_{i,.}^{(l)}$ sont ordonnées dans l'ordre décroissant.

Admettant qu'il y a trois neurones (trois masques) dans la première couche cachée, on dispose des valeurs moyennes suivantes pour le neurone indexé 1 : $\bar{a}_{1,1}^{(1)} = 0.9$, $\bar{a}_{1,2}^{(1)} = 0.7$, et $\bar{a}_{1,3}^{(1)} = 0.8$. Suivant l'étape 3, nous obtenons le classement suivant : $\bar{a}_{1,1}^{(1)}, \bar{a}_{1,3}^{(1)}, \bar{a}_{1,2}^{(1)}$. Dans cet exemple, nous pouvons conclure que le neurone indexé 1 incarne bien le terme GO correspondant, car le rang de son masque cible est 1.

Nous comparons le rang du masque cible des neurones avec le rang des neurones selon leur score de pertinence. Le calcul de ce rang, désigné ci-dessous par rang LRP et décrit dans le bas de la Fig. 4.5, suit les mêmes étapes 1 à 3 en ne considérant pas les masques. Pour chaque exemple x et chaque neurone i dans une couche cachée l, le score $R_i^{(l)}(x)$ est calculé, puis la moyenne qu'on notera $\bar{R}_i^{(l)}$. Par exemple, nous obtenons $\bar{R}_1^{(1)} = 1.9$, $\bar{R}_2^{(1)} = 2.5$, et $\bar{R}_3^{(1)} = 0.3$ respectivement pour les neurones $1, 2$, et 3 de notre exemple précédent. Selon l'étape 3, les scores de pertinence

sont ordonnés de la façon suivante $\bar{R}_2^{(1)}, \bar{R}_1^{(1)}, \bar{R}_3^{(1)}$. Ensuite, sur la base de cette séquence, un rang est attribué à chaque neurone : le neurone 2 obtient le rang 1, et ainsi de suite.

La Fig. 4.6 représente le rang des masques cibles des neurones sur l'axe des ordonnées et leur rang LRP sur l'axe des abscisses. La valeur des rangs est égale au nombre total de neurones, soit 1574. Notons que plus le rang a une valeur faible, plus le neurone représente le concept sous-jacent dans le cas du rang sur les masques cibles ou plus il est important pour la prédiction dans le cas du rang LRP. Sur les données microarray, le rang sur les masques cibles peut avoir une valeur nulle ou une valeur discrète dans l'intervalle [1,16].
Un rang nul signifie que l'activation d'un neurone pour son masque cible est nulle, 603 neurones (38,31 %) sont concernés. Parmi ces neurones, 591 neurones ont une valeur d'activation nulle quel que soit le masque et généralement un rang LRP supérieur à 1000 (colorés en orange dans la Fig. 4.6). Les douze neurones restants sont activés par au moins un autre masque et leur rang LRP est inférieur à 1000 (en vert sur la Fig. 4.6).

971 neurones (61,69 %) ont, quant à eux, une activation positive pour leur masque cible. Ils sont représentés par les points bleus sur la figure. Ils présentent des rangs plus élevés, inférieurs à 1000. Parmi ces neurones, les masques cibles de 850 neurones ont le rang 1, les 121 autres neurones ont un rang compris entre 2 et 16. En conclusion, la plupart des neurones, qui contribuent fortement à la prédiction (rang LRP inférieur à 1000), ont un rang pour leur masque cible proche de 1. Cela signifie que les neurones importants pour la prédiction représentent bien leur terme GO associé.

Concernant les neurones avec un rang nul pour leur masque cible, la majeure partie sont peu importants vis-à-vis de la prédiction. Les termes GO associés peuvent être ignorés. Cependant, les quelques neurones qui ont un rang LRP faible sont beaucoup plus intéressants (colorés en orange dans la Fig. 4.6). Par exemple, le neurone associé au terme "GO :0071644" (*régulation négative de la production de chimiokine (C-C motif) ligand 4*) a un rang LRP de 15, mais il n'est pas activé par son masque cible. Son masque cible est composé de deux gènes, liés par des connexions GO de poids respectif 0,1 et 0,04. À l'inverse, 890 des 1000 premières connexions noGO de la couche d'entrée, qui ont les mêmes valeurs que les connexions GO de norme 0,01, sont entrantes à ce neurone. Comme ces neurones ne sont pas activés par leur masque cible, nous ne pouvons pas conclure qu'ils représentent leur terme GO correspondant. Nous remarquons qu'une grande partie des connexions noGO avec un poids élevé est connectée à ces neurones. De plus, ces connexions noGO connectent principalement des gènes sans annotations GO, n'ayant aucunes connexions GO. Le réseau semble détourner les rôles biologiques sous-jacents que ces neurones sont supposés incarner pour propager l'information provenant de gènes sans annotations GO via les connexions noGO. Ces neurones ne représentent plus leurs termes GO correspondants, mais une information biologique inconnue utile pour les prédictions.

4.2.5 Interprétation biologique des résultats

Dans cette section, nous montrerons comment proposer des interprétations biologiques pertinentes du modèle Deep GONet et de ses prédictions. Nous proposerons trois niveaux d'interprétation :

- global : interprétation générale des prédictions de classe c (dans notre cas le cancer) ;
- intermédiaire : interprétation générale des prédictions de classe c sur un sous-groupe de patients (dans notre cas à l'échelle d'un tissu) ;
- local : explication de la prédiction sur un individu.

Tout d'abord, nous étudierons comment notre modèle détecte le cancer à partir d'échantillons hétérogènes. Ensuite, nous examinerons indépendamment un tissu en extrayant un sous-réseau qui lui est associé à partir des scores de pertinence calculés par LRP. Nous présenterons enfin comment la prédiction individuelle d'un patient peut être expliquée.

a) Interprétation du modèle par rapport à la classe cancer

Dans cette sous-section, nous analyserons le regroupement des patients correctement prédits cancer en fonction de leur profil d'activation. Pour chaque patient, un profil d'activation est constitué de l'activation de tous les neurones du réseau. Pour chacune des couches, nous définissons une matrice d'activation de taille (N, N_l) contenant l'activation de tous les neurones de cette couche pour tous les patients, où N est le nombre de patients et N_l le nombre de neurones dans la couche l. À partir de ces matrices d'activation, nous réalisons une classification hiérarchique ascendante en utilisant le lien moyen et la distance euclidienne comme paramètres. Les dendrogrammes de chaque couche sont représentés sur la Fig. 4.7. Les couleurs sur le dendrogramme représentent les types de tissu présents dans l'échantillon. Dans le dendrogramme de la première couche cachée, on constate que les exemples issus des mêmes tissus ont tendance à être regroupés dans les mêmes partitions. C'est notamment le cas pour les tissus osseux (coloré en orange), sanguins (coloré en rouge), et lymphoïdes (coloré en cyan). Les tissus de même type ont tendance à partager les mêmes profils d'activation, ce qui signifie que certains neurones et leurs termes GO associés sont spécifiques à un tissu particulier. Ce regroupement en fonction du tissu est toujours présent dans la deuxième couche, bien qu'il soit moins significatif. À partir de la troisième couche cachée, le regroupement de patients provenant des mêmes tissus devient de moins en moins visible. De la couche quatre à six, les clusters contiennent des exemples provenant de différents tissus. Cela signifie que les mêmes termes GO sont activés pour la prédiction du cancer, quelles que soient les caractéristiques des entrées (comme la localisation du cancer). Une signature du cancer partagée par tous les tissus a été apprise dans les dernières couches du réseau. En résumé, selon la façon dont notre architecture est structurée, les couches cachées inférieures regroupent des termes GO plus spécifiques. Les neurones associés sont responsables de l'extraction des caractéristiques du cancer propres à un type de tissu. Le modèle devenant progressivement plus général, les couches cachées profondes sont au contraire chargées d'extraire des caractéristiques communes au cancer sur tous les types de tissu. Cela montre que notre classifieur est universel et il est capable d'extraire des caractéristiques partagées par divers tissus cancéreux à travers des termes GO communs. Dans la dernière couche cachée, l'existence de plusieurs clusters indique que différents neurones sont activés pour fournir la même prédiction puisque le signal de la couche d'entrée à la couche de sortie se propage par des chemins différents. Notons que cette capacité à extraire des motifs spécifiques dans les premières couches et des motifs généraux dans les dernières couches est une propriété des modèles d'apprentissage profond, qui a largement été étudiée dans les réseaux de neurones convolutionnels pour l'analyse d'images [RW17]. Grâce à ces profils d'activation, nous voyons comment l'information circule dans le réseau jusqu'à la sortie cancer. Dans les prochaines analyses, nous nous concentrerons sur l'importance des neurones en utilisant le score de pertinence. Il s'agit d'un meilleur indicateur pour des résultats plus précis afin d'évaluer exactement quels neurones contribuent le plus au résultat d'un sous-groupe ou d'un individu [Han+20].

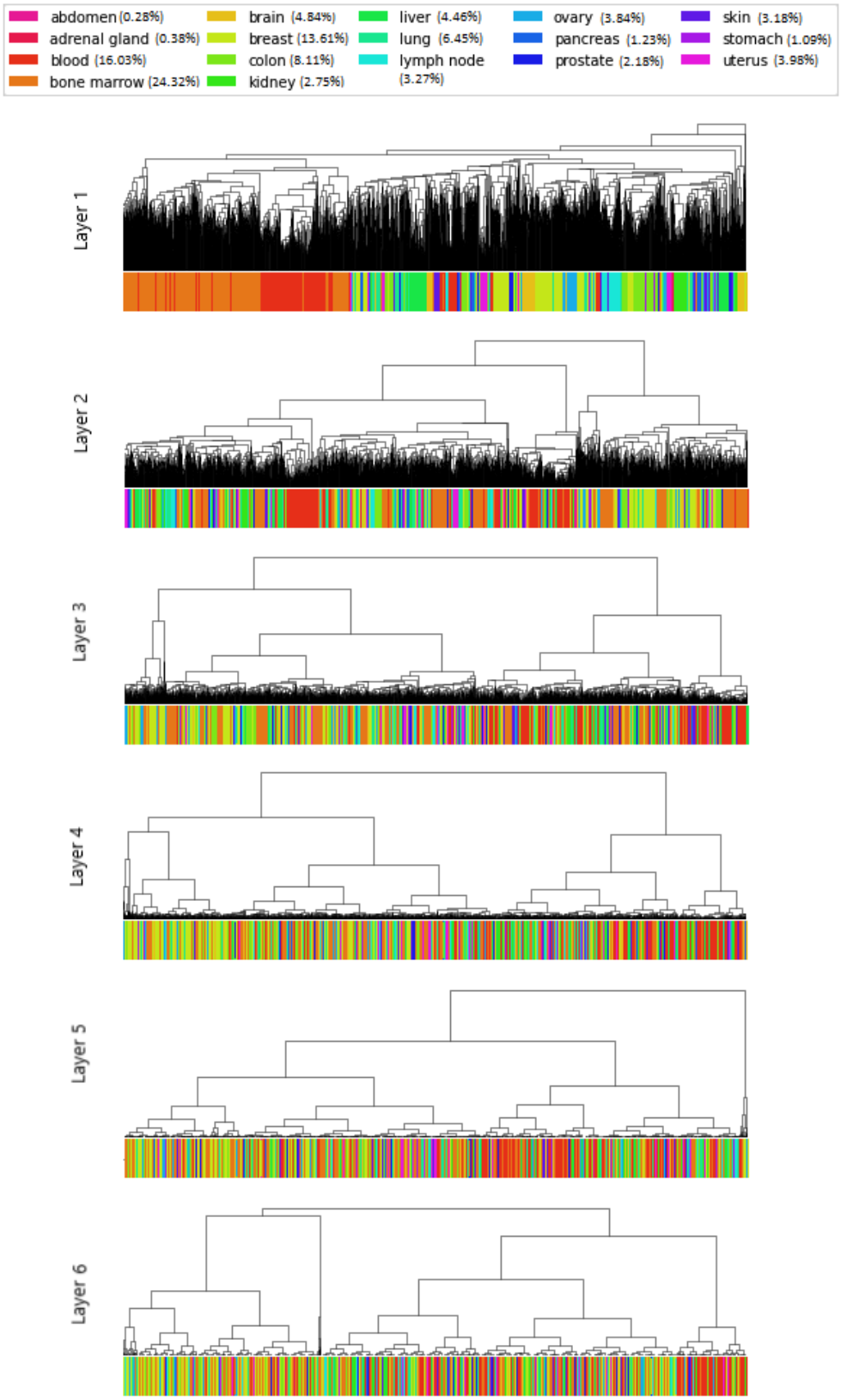


FIGURE 4.7 – *Classification hiérarchique des profils d'activation de chacune des couches du réseau issus des exemples de test correctement prédits cancer.*

b) Interprétation du modèle par rapport à un sous-groupe de patients cancer : zoom sur le tissu mammaire

Dans cette sous-section, nous montrerons comment interpréter biologiquement notre modèle sur un des tissus cancéreux.

Nous proposons un outil qui détecte les principales fonctions biologiques utilisées pour les prédictions et quantifie leur contribution. Comme le montre la Fig. 4.5, nous calculons d'abord le score LRP moyen de chaque neurone sur un échantillon cancer provenant d'un tissu d'intérêt. Ensuite, pour chaque couche, les neurones sont triés en fonction de leur score de pertinence et les plus importants sont retournés avec leur terme GO et leur fonction biologique correspondants. Dans la Fig. 4.8, nous donnons un exemple sur le tissu mammaire où les patients ont été correctement prédits cancer avec une probabilité moyenne de 0,9852. Pour chaque couche cachée, les cinq fonctions biologiques les plus importantes sont rapportées avec leur contribution. Cet outil peut également être utilisé pour déterminer quels gènes sont les plus impliqués dans les prédictions. Il peut aussi être complété par une recherche manuelle dans la littérature afin de valider les liens entre les fonctions retournées et le phénotype prédit. Les experts en biologie et en médecine peuvent donc juger de la pertinence de la prédiction sur la base de cette interprétation.

Parmi tous les termes GO soutenant la prédiction cancer dans ce sous-réseau, en croisant avec la littérature, des liens peuvent être établis avec cette maladie. Dans la première couche cachée, les termes "GO :0015031" et "GO :0006468" liés aux activités protéiques (respectivement de rang LRP 1 et 2) peuvent refléter une perturbation de cette activité. Dans la deuxième couche, les termes "GO :0071420" et "GO :1901258" (respectivement de rang 4 et 5) reflètent une activité immunitaire pouvant faire réponse au cancer [MR10]. Notons que le facteur de stimulation des colonies de macrophages (en anglais *macrophage colony-simulating factor production*), porté par le terme "GO :1901258", est un des facteurs de croissance surexprimés dans de nombreuses tumeurs [CG14]. Deux termes supplémentaires liés à l'activité protéique sont également présents ("GO :0044257" de rang 1 et "GO :0006464" de rang 2). Sur la troisième couche cachée, le terme "GO :0035556", à la 2^{e} place du classement, code pour la fonction biologique transduction du signal intracellulaire relative à la communication cellulaire (en anglais *intracellular signal transduction*). La transduction du signal intracellulaire est une chaîne de réactions biochimiques transmettant des signaux de la surface cellulaire aux récepteurs de divers composants à l'intérieur de la cellule. Cette transduction se termine finalement par une réponse cellulaire sous la forme d'un changement d'état de la cellule comme une croissance ou une différentiation cellulaire. On a découvert que l'hyperactivité de ces voies de signalisation peut augmenter la prolifération des cellules tumorales [SB15]. Dans la quatrième couche cachée, le terme "GO :0042127" de rang 5 peut surligner une prolifération incontrôlée des cellules tumorales. Entre la quatrième et la cinquième couche cachée, différents termes GO font référence à l'activité membranaire ("GO :0071709" et "GO :0055085" de rang 1 et 2, "GO :0044091" de rang 1). De nombreuses altérations des membranes des cellules tumorales ont été détectées, comme la dépolarisation [YB13]. Le terme "GO :0050794" de rang 3 peut indiquer la dérégulation de processus cellulaires. Enfin, concernant le terme "GO :0006739" de rang 3 dans la dernière couche cachée, des études montrent que la quantité de la molécule NADP peut être beaucoup plus importante dans les cellules cancéreuses [CC17]. Notons que si notre objectif avait été de prédire un sous-type de cancer, l'interprétation et les sous-réseaux extraits seraient différents. L'interprétation d'un modèle dépend fortement de la nature de la tâche de prédiction.

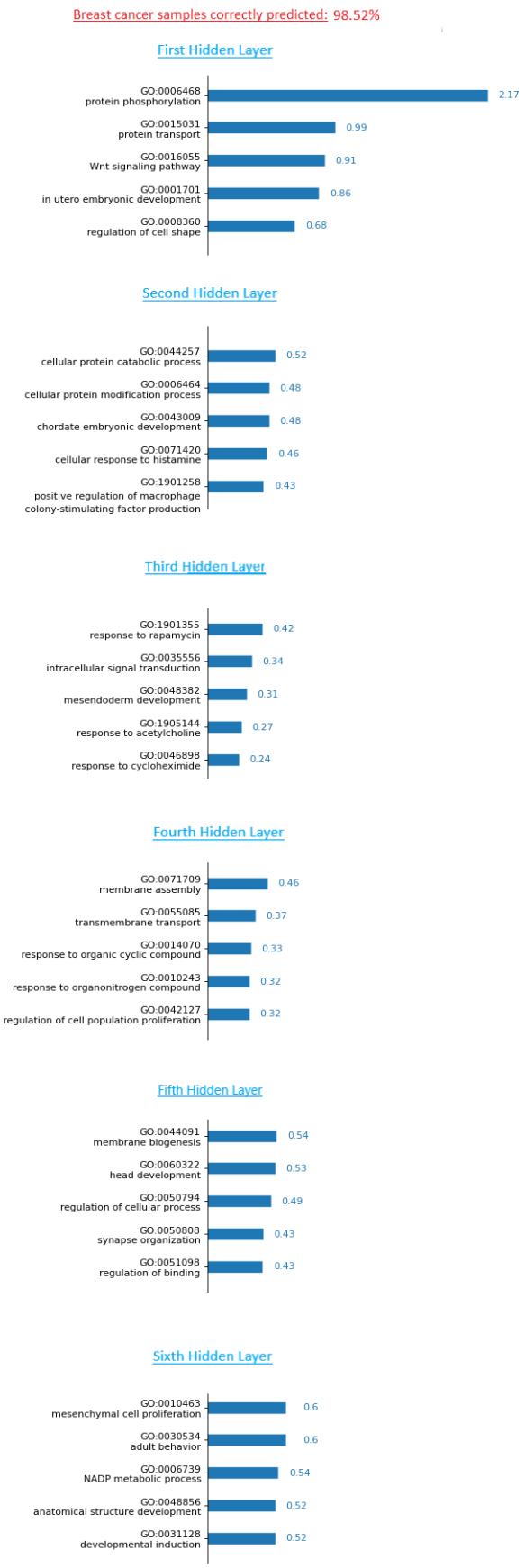


FIGURE 4.8 – ***Interprétation d'un sous-réseau pour le tissu mammaire cancéreux.*** *À chaque couche, les termes GO sont classés en fonction de leur score de pertinence.*

c) Explication d'une prédiction d'un patient

Dans cette sous-section, nous montrerons comment fournir une interprétation biologique d'une prédiction individuelle faite sur un patient pour la rendre compréhensible aux différents utilisateurs (médecin, scientifique...). Le score de pertinence de chaque neurone est ici seulement calculé sur la prédiction d'un patient pour ensuite en déduire un classement de neurones par couche. La Fig. 4.9 présente un exemple d'explication biologique au travers de la prédiction cancer du patient n°24 509, obtenue par Deep GONet avec une probabilité de 0,99. Ce patient fait partie du sous-ensemble précédent relatif au tissu mammaire cancéreux. Comme dans la Fig. 4.8, les cinq neurones les plus importants sont accompagnés de leur score de pertinence.

Dans la première couche cachée de la Fig. 4.9, le terme "GO :0030335" de rang LRP 3 peut mettre en évidence le phénomène de propagation des cellules cancéreuses dans les tissus environnants, qui caractérise le début des métastases tumorales [YWC05]. Dans la deuxième couche cachée, le terme "GO :0010737" de rang 5 codant pour la signalisation de la protéine kinase A (en anglais *protein kinase A signaling*) peut se référer à certaines dysrégulations ou mutations de cette famille protéique contribuant à tous les stades du développement du cancer [Bhu+18]. Dans la troisième couche, le cinquième terme "GO :0048864" peut informer de la production de cellules souches cancéreuses qui ont des caractéristiques similaires aux cellules souches normales. Pour les couches suivantes, nous retrouvons les mêmes termes GO les plus pertinents que dans l'interprétation biologique précédente (cf. : Fig. 4.8). Nous remarquons ainsi qu'il existe des différences sur les termes GO les plus importants entre l'interprétation d'une prédiction individuelle et celle du sous-groupe, surtout dans les premières couches (une à trois). De cette façon, nous pouvons identifier les patients qui ont des caractéristiques distinctes de la moyenne.

Nous avons également observé au cours de nos analyses que pour plus de 99% des patients non-cancer, les termes GO listés dans la Fig. 4.9 sont mal classés. Cela confirme que ces neurones extraient des motifs caractéristiques du cancer en relation avec les fonctions biologiques (prolifération des cellules tumorales, perturbation de l'activité protéique...).
L'explication d'un patient correctement prédit BRCA avec une probabilité de 0,99 peut être trouvée en annexe 7.2.

4.3 Discussion et conclusion

Au travers de cette première approche, nous avons démontré que les performances de prédiction obtenues par Deep GONet sont équivalentes à celles des méthodes opaques d'apprentissage automatique. Contrairement à ses dernières, l'ensemble de l'architecture de Deep GONet est interprétable et facile à comprendre par les biologistes, car elle reflète des connaissances qu'ils ont l'habitude d'employer. Les réseaux de neurones ont en effet la particularité de disposer d'une architecture propice pour intégrer la connaissance. Chaque couche de Deep GONet correspond à un niveau de GO et chaque neurone à un terme de GO. L'ajout d'une régularisation adaptée L_{GO} aide le modèle à mieux respecter ces connaissances en se concentrant sur les connexions réelles entre les objets biologiques. Les expériences présentées sur la détection de cancer montrent comment fournir facilement une interprétation du modèle et de ses prédictions. Dans ce premier travail, l'architecture de Deep GONet est basée sur GO-BP, mais toute autre ontologie structurée comme un DAG, telle que GO-CC, GO-MF ou encore Reactome, peut être implémentée dans le réseau de neurones avec la même approche. Par ailleurs, le modèle peut être appliqué à d'autres jeux de données moléculaires, ou à d'autres tâches de prédiction telles que l'établissement de pronostic, mais cela nécessite un ré-entraînement du modèle. Comme les autres méthodes d'intégration de connaissances, Deep GONet n'est pas tout à fait une méthode *self-explaining* puisqu'un recours à une méthode a posteriori a été nécessaire pour permettre d'affiner l'interprétation. Une étude menée par [Han+20] a montré que l'analyse de la moyenne des poids moyens

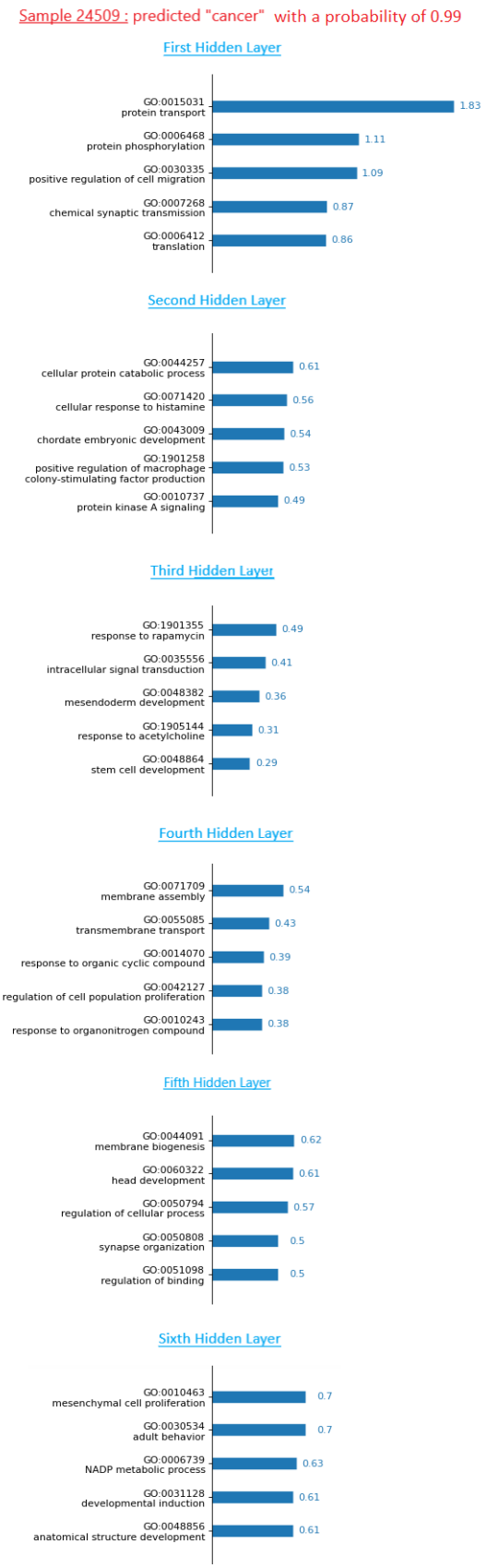


FIGURE 4.9 – ***Explication de la prédiction du patient n°24509.*** *À chaque couche, les termes GO sont classés en fonction de leur score de pertinence.*

des connexions sortantes des neurones (en anglais *weight mean* (WM)) produit des résultats d'interprétation trop généraux et indépendants de la distribution des données. Les auteurs ont mis en évidence que les neurones les plus importants au sens de LRP ne sont pas ceux ayant un score WM élevé. Les variables des données génomiques n'étant pas indépendantes et identiquement distribuées, l'interprétation doit prendre en compte la distribution des données. Dans le cadre de cette étude, nous nous sommes peu attardés sur la significativité du signe des scores de pertinence des neurones, cela méritait plus ample analyse, notamment pour observer les différences entre un profil cancer et un profil non-cancer. De même, nous pourrions retracer les chemins de propagation les plus empruntés et identifier à quelle catégorie ils appartiennent (GO ou noGO). Cela permettrait de contribuer à la définition des sous-réseaux de propagation, par exemple par type de tissu.

Nous rappelons que le but de l'interprétation est d'expliquer comment le modèle fonctionne et non comment la biologie fonctionne. Parfois, il n'y a pas de relations évidentes entre les fonctions biologiques, retournées par l'explication et le phénotype prédit. Cela ne signifie pas nécessairement que les prédictions ne sont pas fiables. De plus, un modèle recherche des corrélations entre la sortie et l'entrée et non des causalités. Lorsqu'une fonction biologique, qui ne semble pas liée au phénotype, est retournée, il est possible que cette fonction ait une corrélation indirecte ou soit liée par une relation de causalité inconnue avec le phénotype. Cependant, plus les fonctions biologiques retournées par l'interprétation sont cohérentes avec le phénotype, plus nous pouvons faire confiance aux prédictions du modèle. Si la majeure partie de l'explication est incohérente avec les connaissances biologiques actuelles, la fiabilité du modèle doit être remise en question. Le modèle peut être surajusté ou induit en erreur par un biais dans l'ensemble d'apprentissage.

Bien que l'interprétation du modèle ne soit pas un outil de découverte biologique, certaines parties de notre réseau de neurones pourraient être étudiées de cette manière. Nous nous référons, en particulier, aux connexions noGO à poids élevé et aux neurones détournés de leur terme GO. Ces éléments connectent au réseau les gènes non annotés. Il pourrait être intéressant de comprendre pourquoi ces gènes ont été utilisées pour la prédiction, ils devraient être liés au phénotype. Nous pourrions également étudier comment l'expression de ces gènes est combinée dans les couches cachées. Les gènes annotés connectés au même neurone pourraient avoir des fonctions biologiques proches liées au phénotype prédit. Notre modèle peut donc contribuer à enrichir GO en soulevant de nouvelles hypothèses qui doivent être validées par des expériences biologiques.

Les modèles de l'état de l'art qui ont basé l'architecture du réseau de neurones sur GO sont peu nombreux et le font de manière différente. Dans GONN [PWS19], premièrement, comme mentionné dans l'état de l'art, seule une partie des neurones d'une couche cachée sur les deux couches cachées du réseau de neurones est associée à des termes GO des sous-ontologies GO-BP et GO-MF. Il demeure quelques incertitudes sur le traitement des connaissances et la définition de l'architecture du réseau. Tout d'abord, bien que les auteurs aient déterminé le nombre de termes GO par couche, ils n'ont pas précisé comment ils avaient réussi à déterminer le nombre de niveaux. Ils ont probablement utilisé la plus courte distance à la racine du fait qu'ils en trouvent 12 au lieu de 20 pour BP. Dans ce cas, pour un même niveau donné, on peut se retrouver avec des termes GO ayant des degrés de spécification différente. Cela pourrait perturber le modèle d'apprentissage profond dans le sens où celui-ci extrait des caractéristiques spécifiques dans les premières couches du réseau et plus générales dans les dernières couches. Néanmoins, sans prise en compte de la hiérarchie, cela peut ne pas avoir d'impacts dans le fonctionnement du réseau. Les auteurs ont choisi le troisième niveau des deux sous-ontologies BP et MF, car c'est celui qui donnait a priori de meilleures performances. À partir de là, ils ont supprimé un certain nombre de termes, soit parce qu'ils sont redondants, c'est-à-dire qu'ils annotent les mêmes ensembles de gènes, soit ils annotent peu de gènes. Après ce filtrage, les auteurs indiquent un nombre de 854 termes GO finalement retenus. Ensuite, bien qu'ils montrent dans la représentation de leur modèle

que les gènes non annotés ont été connectés à une centaine de neurones supplémentaires sur la même couche cachée que celle sur laquelle les termes GO ont été ajoutés. À aucun moment, ce détail est discuté ou analysé alors que cela s'avère un hyperparamètre important. Contrairement à Deep GONet, où ces gènes sont directement connectés aux neurones existants, ceux-là n'ont aucune signification biologique particulière. Il est alors difficile de pouvoir formuler des hypothèses sur une caractérisation biologique de ces gènes en s'aidant du modèle. Au contraire, dans Deep GONet, si un neurone associé à un terme GO a un score de pertinence élevé avec des gènes annotés et non annotés, cela pourrait révéler que ces gènes non annotés ont des caractéristiques biologiques proches des gènes annotés. On pourrait ainsi en déduire de nouvelles annotations GO. Il faut également préciser que ces neurones additionnels dans GONN sont également connectés aux gènes annotés, le rapport entre les deux types de connexions n'est pas donné et il est aussi probable que le signal ne passe que par ces neurones. Une régularisation L2 est appliquée sur les matrices de poids de toutes connexions confondus, ne forçant donc pas le signal à passer par les connexions reflétant des relations réelles, contrairement à la régularisation adaptée L_{GO} mise en place dans Deep GONet. Les quelques analyses conduites après apprentissage du modèle consistent principalement à mettre en évidence que leur modèle performe mieux que les modèles existants opaques. Du côté de l'interprétation, ils proposent une interprétation globale du modèle en identifiant les termes GO les plus importants pour un phénotype, en multipliant les matrices de poids. Ils ont également vérifié dans la littérature le lien avec ce phénotype. Nous disposons seulement d'une liste qui n'est pas accompagnée du score obtenu, de ce fait, il n'est pas clair si les neurones additionnels peuvent être aussi, voire plus importants. Comme discuté en amont, l'analyse des poids n'est pas suffisante, la distribution des données d'entrée ainsi que la valeur d'activation des neurones des couches cachées à la couche de sortie est également un facteur pouvant influencer la pertinence d'un neurone vis-à-vis de la prédiction.

L'autre approche évoquée dans l'état de l'art est ParsVNN [Hua+21], publiée après Deep GONet. Même si plusieurs niveaux de l'ontologie GO-BP (0 à 5) ont été intégrés dans ce modèle, l'objectif visé reste différent puisque les auteurs cherchent à simuler la biologie au sein du réseau de neurones alors que le nôtre est de contraindre le modèle par la connaissance. De ce fait, un terme GO est représenté non pas par un neurone, mais par un ensemble de neurones. De cette manière, seuls 2046 termes des six premiers niveaux de GO ont pu être représentés. Les gènes non annotés ne sont pas inclus, seuls 3008 gènes annotés sont intégrés dans le modèle. La découverte biologique s'en trouve donc plus limitée. Le second objectif de cette approche est de déterminer un sous-réseau par type de cancer au moyen d'une technique d'élagage. Les sous-réseaux obtenus contiennent de 12 à 34 termes GO répartis sur quatre à six niveaux. L'interprétation se fait au niveau global et consiste principalement à analyser les sous-réseaux élagués.

Les travaux similaires intégrant une hiérarchie de connaissances sont postérieures aux nôtres. Ces travaux intègrent plusieurs niveaux à tous les niveaux de l'ontologie Reactome sans nécessairement le justifier. Les gènes non annotés ne sont pas considérés et l'interprétation est essentiellement réalisée au niveau global. Deep GONet se distingue donc de l'état de l'art sur plusieurs points. L'architecture profonde est totalement interprétable et intègre des gènes non annotés au moyen d'un terme de régularisation personnalisé permettant de faciliter la découverte biologique. Des explications intelligibles sont fournies pour accompagner les prédictions individuelles.

À ce premier travail, différentes pistes d'amélioration peuvent être envisagées. Tout d'abord, au sujet de l'architecture de Deep GONet, les relations entre les gènes et les termes GO des couches cachées profondes, de même que les relations inter-niveaux (entre deux niveaux GO non adjacents), ne sont pas représentées, même si une partie de l'information propagée dans le réseau au travers des connexions GO peut permettre d'imiter ces relations. Pour mieux les représenter, il est possible de mettre en place des connexions qui sauteraient certaines couches dans le réseau. Ces connexions sont généralement désignées de connexions résiduelles dans la littérature [He+16]. Elles ont l'avantage de préserver une bonne rétro-propagation du gradient. À notre connaissance,

cette proposition a été jusqu'à présent peu explorée dans l'état de l'art [Hua+21]. Nous pourrions également présélectionner des termes GO plus adaptés au problème à résoudre. Nous pourrions aussi enrichir l'interprétation à partir d'autres connaissances. En effet, les biologistes et bioinformaticiens ont l'habitude de travailler avec différentes bases de connaissances complémentaires, chacune abordant la biologie sous un angle différent. Malgré la richesse de l'ontologie GO-BP, le recours seulement à celle-ci peut s'avérer pas assez suffisant pour fournir des explications biologiques complètes des prédictions. Inclure différentes connaissances pourrait permettre aussi de contrebalancer le fait que ces bases sont en constante évolution et peuvent contenir des erreurs. De plus, elles ne caractérisent pas forcément les mêmes gènes et pourraient se complémenter sur ce point. Différentes solutions peuvent être proposées pour inclure d'autres connaissances, soit en conduisant un enrichissement a posteriori comme dans [Han+20] ne nécessitant pas le réapprentissage du réseau, soit en ajoutant une deuxième branche à Deep GONet qui représenterait par exemple les voies métaboliques de l'ontologie Reactome. Cette seconde solution demande un réapprentissage, mais permet de conserver la propriété de modèle interprétable par construction. Une concaténation à la manière de GONN, où certains neurones d'une même couche cachée seraient associées à des objets biologiques d'une première base de connaissances et d'autres à une seconde base, pourrait être envisagée à condition que les niveaux extraits aient de profondeurs similaires. La solution proposée par BDKANN de concaténation de différentes couches de connaissances est moins envisageable dans le cadre de l'intégration de hiérarchies de connaissances. Concernant la régularisation mis en place, nous pourrions envisager une régularisation spectrale fondée sur le laplacien [Ton+20] ou sur le *graph lasso* [JOV09] pour mieux tenir compte de la topologie du graphe. De la même façon que ParsVNN, nous pourrions aussi envisager d'élaguer Deep GONet en nous basant sur les scores de pertinence obtenus avec LRP. Des premiers travaux existent déjà dans ce sens [Yeo+21 ; Yu+18b]. Enfin, nous pourrions automatiser la recherche de liens entre les termes GO retournés et le phénotype prédit dans la littérature biologique pour valider les explications obtenues. Nous pourrions recourir aux techniques de traitement automatique de la langue basées sur l'apprentissage profond. Il existe déjà des outils tels que LISC - Literature Scanner [Don19].

Chap.

5 GraphGONet

Contenu du chapitre

Par ce second travail, nous souhaitions proposer une solution à la première piste d'améliorations, évoquée au sujet de Deep GONet, qui concerne la valorisation plus complète de la sémantique offerte par le graphe de connaissances. Pour ce faire, nous avons choisi de recourir cette fois-ci à la fois au réseau FFNN et au réseau GNN, afin de considérer au maximum l'intégralité de la base de connaissances et sa sémantique, sans un surcoût de paramètres à apprendre. Tout comme Deep GONet, le principal objectif reste l'interprétation qui ne doit pas se faire au détriment de la capacité de prédiction du modèle. Les explications restent compréhensibles aux utilisateurs et ont l'avantage d'être rendues automatiquement, faisant de cette nouvelle approche une méthode *self-explaining*. Ce travail a fait l'objet d'une publication [1] dans le journal Bioinformatics [BZH22a], ainsi qu'une présentation orale lors de la conférence nationale de bio-informatique JOBIM2022 [BZH22b].

Dans un premier temps, la nouvelle méthode, intitulée GraphGONet, sera décrite. Puis, nous présenterons les résultats obtenus sur des jeux de données réelles. Enfin, nous discuterons de ce second travail et des perspectives.

5.1 Méthode

Dans ce qui suit, nous allons présenter GraphGONet, un nouveau réseau de neurones interprétable par construction, dont la couche d'entrée correspond au profil GE d'un patient et les couches cachées intègrent les connaissances issues de GO. La méthode est illustrée dans la Fig. 5.1 et montre que le signal partant de la couche d'entrée des gènes est propagé séquentiellement à travers les couches GO. Le signal passe ensuite par une couche de sélection, où il est concaténé et masqué pour réaliser finalement la tâche de prédiction dans la couche de sortie. Une description complète de la méthode est fournie dans la sous-section suivante 5.1.1. L'acquisition d'une explication est détaillée dans la sous-section 5.1.2.

1. Code disponible sur https://forge.ibisc.univ-evry.fr/vbourgeais/GraphGONet.

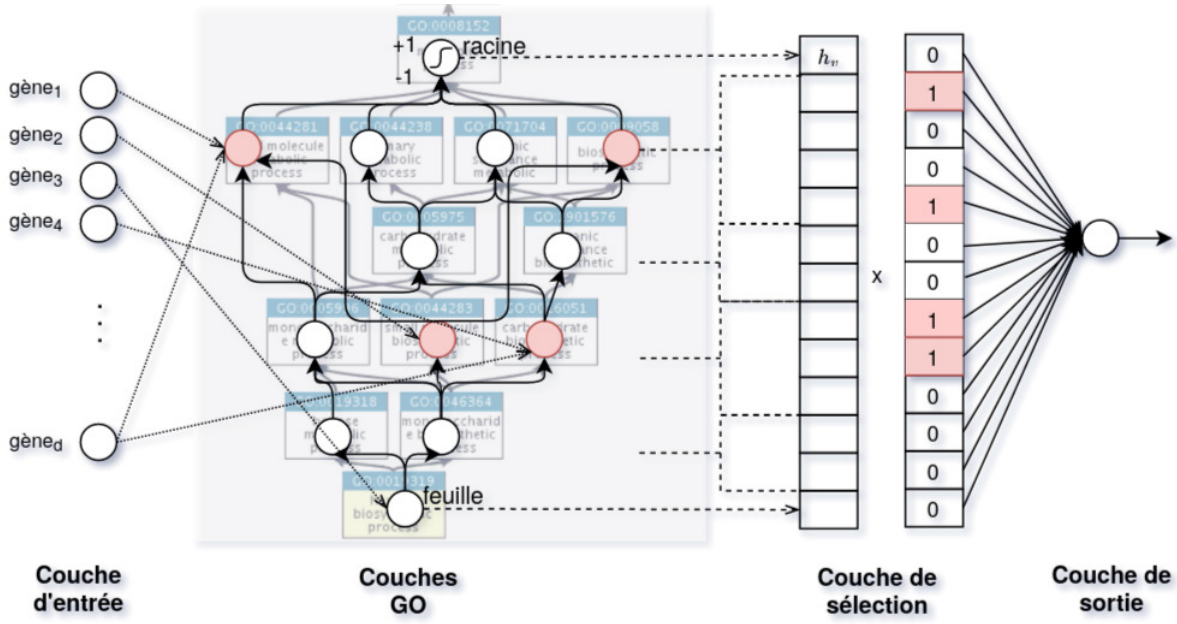


FIGURE 5.1 – ***Illustration de GraphGONet.*** *Les neurones de la couche d'entrée reçoivent le signal des gènes. Les flèches en pointillés correspondent aux annotations entre les gènes et les termes GO qui sont représentés par les neurones dans les couches cachées. Les relations entre les termes GO sont représentées par des flèches pleines. Les flèches en tirets représentent la concaténation des valeurs d'activation des neurones. La couche de sélection résulte des opérations de concaténation et de masquage.*

5.1.1 Description de l'architecture

Soit (X, Y) un exemple d'apprentissage, où $X = [x_1, ..., x_d]$ est le profil d'expression d'un patient avec d le nombre de gènes et $Y = \{0, 1\}^C$ l'indicateur de classe à prédire avec C le nombre de classes. $y_c = 1$ si le patient appartient à la classe c sinon $y_c = 0$. Chaque exemple n'appartient qu'à une seule classe. Un neurone de la couche d'entrée reçoit l'expression d'un gène. Cette couche d'entrée est connectée à un ensemble de nœuds organisés en couches, qui imite la structure de GO. Chaque couche de la hiérarchie représente un niveau de GO où la première couche cachée correspond au niveau le plus spécifique et la dernière couche cachée représente une racine de l'ontologie. Chaque neurone de ces couches représente un terme GO et chaque connexion entre deux neurones représente une relation entre deux termes GO. Les connexions sont orientées des niveaux GO inférieurs vers les niveaux GO supérieurs. Deux neurones, représentant des termes GO non liés, ne sont pas connectés. Il existe également des connexions résiduelles puisque les termes GO de niveaux non adjacents peuvent être liés. Nous définissons par $G(v)$ l'ensemble des gènes associés à un terme GO correspondant à un neurone v dans GraphGONet et par $\mathcal{N}(v)$ l'ensemble de neurones correspondant aux termes GO fils du neurone v. L'expression génétique est propagée aux neurones des couches cachées par des connexions qui représentent des annotations entre les termes GO et les gènes. Un neurone v n'est connecté qu'aux gènes de $G(v)$. La valeur d'activation d'un neurone h_v est calculée à partir du vecteur d'expression $X_{G(v)}$ restreint aux gènes dans l'ensemble $G(v)$ et l'activation des neurones fils dans l'ensemble $\mathcal{N}(v)$. L'activation h_v du neurone v se définit de la manière suivante :

$$h_v = \begin{cases} \sigma\left(w_G h_{G(v)} + w_{\mathcal{N}} h_{\mathcal{N}(v)}\right) & \text{si } |\mathcal{N}(v)| > 0 \\ \sigma\left(h_{G(v)}\right) & \text{si } |\mathcal{N}(v)| = 0 \end{cases} \tag{5.1}$$

où $w_G, w_{\mathcal{N}} \in \mathbb{R}$ sont des paramètres appris partagés par tous les neurones v, $|\cdot|$ désigne la cardinalité, σ est la fonction d'activation tanh telle que $\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$. Dans notre

modèle, le choix de la fonction tanh est plus pertinent que la fonction ReLU. La fonction tanh sature les neurones sélectionnés par la couche de sélection, à des valeurs +1 ou -1, ce qui permettra de faciliter l'interprétation. $h_{G(v)}$ et $h_{\mathcal{N}(v)}$ correspondent respectivement à la combinaison linéaire de l'expression de l'ensemble $G(v)$ (Eq. (5.2)) et l'activation moyenne de l'ensemble $\mathcal{N}(v)$ (Eq. (5.3)) :

$$h_{G(v)} = W_v X_{G(v)} + b_v \tag{5.2}$$

où $(W_v \in \mathbb{R}^{|G(v)|}, b_v \in \mathbb{R})$ sont des paramètres à apprendre.

$$h_{\mathcal{N}(v)} = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} h_u \tag{5.3}$$

L'activation des neurones est calculée de manière séquentielle à partir des neurones les plus spécifiques jusqu'à la racine. Cependant, les neurones ayant l'information sur leur voisinage disponible au même moment sont traités simultanément. Les paramètres (W_v, b_v) des connexions entre un neurone et ses gènes associés sont spécifiques, tandis que les paramètres $(w_G, w_{\mathcal{N}})$ propageant les activations au travers des couches GO sont communs à tous les neurones. Comparé à un FFNN (tel qu'un MLP), ce partage de paramètres, inspiré des GNNs, réduit fortement le nombre de paramètres à apprendre.

La partie suivante du modèle consiste à sélectionner les neurones les plus activés en valeur absolue. Les termes GO qui leur sont associés seront utilisés comme support d'explication des prédictions. Le processus consiste à (1) concaténer tous les neurones des couches cachées GO tel que $H_{concat} = \text{CONCAT}(h_v | \forall v)$, (2) calculer un masque vectoriel M permettant d'identifier les neurones les plus activés, ainsi $m_v = 1$ si $v \in \text{top}(r)$, $m_v = 0$ sinon, où r est le ratio de sélection et top la fonction retournant les indices des $\lceil |\mathcal{V}| r \rceil$ neurones choisis ($|\mathcal{V}|$ représente le nombre de termes GO), (3) appliquer le masque $H_{select} = H_{concat} \cdot M$. Notons que r est un hyperparamètre du modèle à déterminer.

La dernière couche est composée d'un neurone pour chacune des C classes et résulte d'une combinaison linéaire des neurones de H_{select}. La valeur d'activation d'un neurone de sortie est calculée à partir de $z_c = \sum_{j=1}^{|\mathcal{V}|} h_{select,j} w_{j,c} + b_c$, où $(W \in \mathbb{R}^{|\mathcal{V}| \times C}, b \in \mathbb{R}^C)$ sont des paramètres à apprendre. Les activations de sortie sont transformées en probabilités en utilisant la fonction softmax telle que

$$o_c = \frac{\exp(z_c)}{\sum_{j=1}^{C} \exp(z_j)} \tag{5.4}$$

Le modèle est entraîné de bout en bout avec la descente de gradient habituelle en recherchant les paramètres $(W_v, b_v, w_G, w_{\mathcal{N}}, W, b)$ pour minimiser la fonction de coût suivante :

$$\mathcal{L} = \sum_{c=1}^{C} (-y_c \log o_c) \tag{5.5}$$

GraphGONet peut être vu comme la combinaison d'un type particulier de FFNN et de GNN. En effet, dans notre proposition de FFNN, la couche d'entrée peut être connectée à chaque couche cachée et toutes les couches cachées sont connectées à la couche de sortie à travers une couche de sélection. La propagation du signal à travers les couches cachées qui représentent la hiérarchie de l'ontologie GO est inspirée des lois de propagation des GNNs.

5.1.2 Interprétation du modèle

Notre modèle fournit automatiquement une prédiction et une explication pour un patient donné. L'explication prend la forme d'une liste de termes GO impliqués dans le calcul final de la prédiction, avec leur score d'importance. Le nombre de termes GO dans la liste est déterminé par le ratio de sélection r. Par exemple, quatre termes GO sont choisis dans la Fig. 5.1. L'importance

d'un terme GO relativement à une classe c dépend du poids de la connexion entre son neurone associé dans la couche de sélection et le neurone représentant la classe c dans la couche de sortie. Par conséquent, nous réutiliserons le score d'importance comme métrique d'interprétation, qui calculera ici la proportion du signal de sortie passant par les neurones de H_{select} et leurs connexions sortantes. Basé sur l'analyse de saillance, le score de pertinence R_j^c d'un terme GO j est donné par :

$$R_j^c = h_{select,j} \times w_{j,c} \tag{5.6}$$

Ce score peut être vu comme le résultat de l'application de la méthode *Gradients× Inputs* (GI) [SVZ14] par rapport à un neurone de la couche de sélection. La formule générale de cette méthode peut être retrouvée dans la Table 4.1 page 47. Les termes GO non impliqués dans la prédiction finale auront leur score fixé à zéro puisque le masque annule leur activation. Il ne s'agira pas du même ensemble de termes GO sélectionnés pour chaque patient.

5.2 Résultats

L'approche a été évaluée sur les trois jeux de données à disposition : microarray, TCGA, et la cohorte d'Oncodesign (décrit à la sous-section 2.2.4).

5.2.1 Choix des couches GO

Dans les expériences suivantes, nous avons basé les couches cachées de GraphGONet uniquement sur la sous-ontologie GO-BP. Celle-ci a été choisie pour les mêmes raisons qu'évoquées précédemment pour Deep GONet. Cependant, comme pour Deep GONet, il est tout à fait possible de la remplacer par les sous-ontologies GO-MF ou GO-CC.

La version de GO-BP utilisée dans cette approche date du 01-06-2020 et contient à l'origine 29 112 termes GO. Pour chaque jeu de données, un graphe GO est constitué des termes GO annotés avec l'ensemble des gènes d'entrée. 67,37 % des gènes de microarray (resp. 32,56 % pour TCGA et 31,29 % pour la cohorte d'Oncodesign) sont liés à au moins un terme GO-BP. Contrairement à Deep GONet, les gènes qui ne sont associés à aucun terme GO sont supprimés. Cependant, nous conservons les gènes liés uniquement à la racine. On s'attend à ce qu'ils détiennent un rôle dans la programmation de processus biologiques, mais rien au moment de l'annotation n'a pu le confirmer avec certitude. Si un gène est annoté avec un terme GO et les ancêtres de ce terme GO, nous ne considérons pas les annotations avec les ancêtres. Sur la base de la loi de propagation de GraphGONet, l'information sera diffusée aux ancêtres. Les feuilles et les termes GO dont les descendants n'ont pas de liens avec les gènes sont supprimés. En fonction du matériel informatique utilisé, les graphes GO entiers pouvaient ne pas tenir en mémoire du fait de l'encodage des connexions entre les gènes et les termes GO sous forme de masque. Nous avons dû dans ces cas-là couper le premier niveau de feuilles de chaque graphe. Selon le principe de transitivité, un terme GO parent hérite de l'ensemble des gènes de ses enfants, nous pouvons donc connecter les parents des feuilles supprimées aux gènes avec lesquelles celles-ci étaient annotées. Il en résulte un graphe GO à 19 niveaux où le premier niveau de feuilles a été élagué avec respectivement ($|\mathcal{V}| = 10\ 663$, $|\mathcal{E}| = 23\ 909$) pour microarray et ($|\mathcal{V}| = 10\ 636$, $|\mathcal{E}| = 23\ 824$) pour TCGA et un graphe GO à 20 niveaux avec ($|\mathcal{V}| = 15\ 849$, $|\mathcal{E}| = 36\ 116$) pour les données d'Oncodesign. La Table 5.1a (resp. la Table 5.1b et la Table 5.1c) reporte par niveau GO-BP le nombre de termes GO, leur degré moyen, la moyenne et l'écart-type du nombre d'annotations avec les gènes sur le jeu de données microarray (resp. TCGA et celui d'Oncodesign). Si l'on ne considère pas le premier niveau contenant uniquement la racine, le deuxième niveau pour la cohorte (resp. troisième niveau pour TCGA et microarray) est le plus annoté en moyenne.

niveau GO	1	2	3	4	5	6	7	8	9
#termes GO	1	23	116	294	603	1089	1404	1500	1614
avg(d^+)	0,00	1,00	1,21	1,36	1,74	1,93	2,03	2,19	2,33
avg(d^-)	28,00	11,65	7,08	6,36	5,33	3,25	2,43	2,22	1,75
avg(gènes_annotés)	372	31,10	84,41	51	35,12	36,70	36,01	31,61	27,61
std(gènes_annotés)	-	37,49	175,67	110,81	71,99	82,89	69,06	75,15	50,75

niveau GO	10	11	12	13	14	15	16	17	18	19
#termes GO	1453	1099	706	388	198	96	53	20	5	1
avg(d^+)	2,46	2,55	2,60	2,58	2,77	2,98	3,04	3,25	3,40	2,00
avg(d^-)	1,46	1,12	0,94	0,82	0,80	0,94	0,58	0,45	0,20	0,00
avg(gènes_annotés)	26,78	26,20	31,29	47,13	25,98	21,70	11,52	12,27	16,67	6,00
std(gènes_annotés)	62,12	53,46	105,56	220,48	53,06	23,66	17,81	17,51	3,21	-

(a)

niveau GO	1	2	3	4	5	6	7	8	9
#termes GO	1	23	117	295	604	1085	1403	1495	1577
avg(d^+)	0,00	1,00	1,21	1,36	1,75	1,93	2,02	2,18	2,32
avg(d^-)	28,00	11,70	7,03	6,33	5,30	3,25	2,41	2,16	1,77
avg(gènes_annotés)	194	15,56	36,19	24,41	15,20	15,77	15,26	13,27	11,45
std(gènes_annotés)	-	19,56	77,82	56,79	33,12	44,52	30,21	33,83	20,93

niveau GO	10	11	12	13	14	15	16	17	18	19
#termes GO	1443	1101	706	395	208	102	55	20	5	1
avg(d^+)	2,46	2,57	2,58	2,57	2,78	3,00	3,07	3,25	3,40	2,00
avg(d^-)	1,49	1,12	0,96	0,86	0,81	0,91	0,56	0,45	0,20	0,00
avg(gènes_annotés)	10,68	11,16	12,59	18,61	10,24	7,97	4,31	5,45	6,33	3,00
std(gènes_annotés)	25,96	25,87	41,63	87,46	21,33	9,13	6,47	8,32	2,08	0,00

(b)

niveau GO	1	2	3	4	5	6	7	8	9	10
#termes GO	1	24	129	344	741	1312	1825	2129	2379	2263
avg(d^+)	0,00	1,00	1,16	1,33	1,67	1,92	2,04	2,15	2,32	2,48
avg(d^-)	30,00	13,04	7,60	6,93	5,55	3,71	2,81	2,39	1,94	1,60
avg(gènes_annotés)	202	51,80	24,99	17,60	12,00	12,79	11,95	10,33	8,73	7,94
std(gènes_annotés)	-	119,44	51,09	43,17	29,52	38,23	26,74	28,77	16,36	19,96

niveau GO	11	12	13	14	15	16	17	18	19	20
#termes GO	1830	1299	758	411	216	101	52	22	10	3
avg(d^+)	2,60	2,58	2,58	2,62	2,73	3,04	3,15	2,95	3,20	3,33
avg(d^-)	1,25	0,99	0,91	0,92	0,85	0,79	0,71	0,86	0,50	0,00
avg(gènes_annotés)	7,27	7,36	8,67	6,84	4,95	4,58	4,00	3,00	6,22	2,67
std(gènes_annotés)	17,68	29,52	52,89	15,53	5,97	5,58	5,99	2,99	5,97	1,53

(c)

TABLE 5.1 – ***Détails des niveaux GO-BP représentés et leur connectivité avec les gènes sur (a) le jeu de données microarray, (b) le jeu de données TCGA et (c) le jeu de données d'Oncodesign.*** *Les première, deuxième et troisième lignes indiquent le nombre de termes GO, le degré entrant (d^-) et sortant (d^+) moyen (avg) par niveau de la sous-ontologie GO-BP, les quatrième et cinquième lignes la moyenne (avg) et l'écart-type (std) du nombre d'annotations avec les gènes.*

5.2.2 Analyse de sensibilité

Nous avons mené deux expériences pour évaluer l'efficacité de GraphGONet par rapport à l'état de l'art. GraphGONet a été entraîné par l'optimiseur Adam avec un pas d'apprentissage adaptatif de 0,001 et une taille de batch de 64. L'arrêt prématuré a été employé avec une patience de 5 et un delta de 0,001. Nous avons effectué sur le jeu de données microarray et la cohorte une classification binaire avec un seul neurone de sortie muni de la fonction sigmoid et sur TCGA une classification multiclasses avec un neurone de sortie par classe dont l'activation finale est calculée au moyen de la fonction softmax. Les mesures de performance indiquées dans les Figures 5.2a-d ci-dessous sont estimées sur les ensembles de test pour microarray et TCGA. Sur la cohorte, nous avons dû procéder à une validation croisée du fait que le jeu de données contienne moins de 250 patients. Les performances ont donc été estimées sur chaque bloc de validation. Les expériences décrites ont été exécutées sur un GPU NVIDIA RTX 2080Ti et un NVIDIA Tesla® V100 GPU 32G/512GB (pour la cohorte d'Oncodesign exclusivement) en utilisant PyTorch v1.7.1 et PyTorch Geometric v1.6.3. Les scores de contribution ont été calculés par la bibliothèque Python Captum v0.3.1.

Dans la première expérience, nous analyserons la couche de sélection pour mesurer son rôle dans GraphGONet. Cette couche est un module clé pour rendre le modèle *self-explaining*. Elle extrait un sous-ensemble des neurones les plus informatifs et leurs termes GO associés pour réaliser la tâche de prédiction. Ce sous-ensemble peut être directement utilisé comme support de l'explication de la prédiction. Le ratio de sélection r détermine la taille de ce sous-ensemble. Un modèle est appris avec un ratio fixe et ce même ratio sera utilisé en phase d'inférence. Comme décrit dans la section Méthodes 5.1.1, la nomination des termes GO les plus informatifs est réalisée par la couche de sélection de GraphGONet. Ce choix se base sur la valeur absolue d'activation de leurs neurones associés. De cette façon, nous avons évalué le processus de sélection et la valeur de l'hyperparamètre r et avons comparé ce processus à une sélection aléatoire. Nous faisons varier la valeur de r dans un intervalle de 0,00005 à 1. Lorsque $r = 1$, tous les termes GO sont sélectionnés. Dix modèles sont appris pour chaque valeur de r avec une initialisation aléatoire et différente des poids et des biais.

La moyenne et l'écart-type de l'accuracy des modèles sont indiqués en fonction de la valeur de r dans la Fig. 5.2a pour le jeu de données de microarray, la Fig. 5.2b sur le jeu de données TCGA et la Fig. 5.3 sur le jeu de données d'Oncodesign. D'autres métriques mesurant les performances, l'AUPRC et le F1-Score sont présentées en annexe Figures 7.3 et 7.4. Dans ce qui suit, nous commenterons exclusivement la courbe d'accuracy des modèles, mais nous pouvons observer les mêmes tendances avec les autres métriques.

On constate tout d'abord qu'en général, la sélection aléatoire est moins performante que la sélection "top" sur les trois jeux de données. Sur le jeu de données microarray, les performances avec la sélection aléatoire commencent à diminuer progressivement de 0,934 à $r = 0,5$ pour atteindre à $r = 0,00005$ une accuracy de 0,661, correspondant à la proportion de la classe majoritaire. À l'inverse, avec la sélection "top", les performances augmentent de 0,937 à 0,945 avec des valeurs de sélection de 1 à 0,1, puis diminuent légèrement de 0,945 à 0,919 avec un ratio compris entre 0,1 et 0,0005. L'application de la sélection "top" peut donc aider non seulement à interpréter, mais aussi à améliorer les performances. En gardant la totalité ou la moitié des neurones, les performances sont moins bonnes qu'avec une sélection avec des ratios plus petits. Les performances chutent finalement pour atteindre une accuracy de 0,744 à $r = 0,00005$. Sur le jeu de données TCGA, nous avons des résultats similaires, sauf que la sélection aléatoire avec des ratios de 0,5 à 0,01 est aussi performante que la sélection "top".

Sur le jeu de données d'Oncodesign, malgré une forte variabilité causée par la validation croisée sur un jeu de données de petite taille, nous retrouvons un comportement très proche des courbes obtenues sur les données microarray. Les performances avec la sélection aléatoire

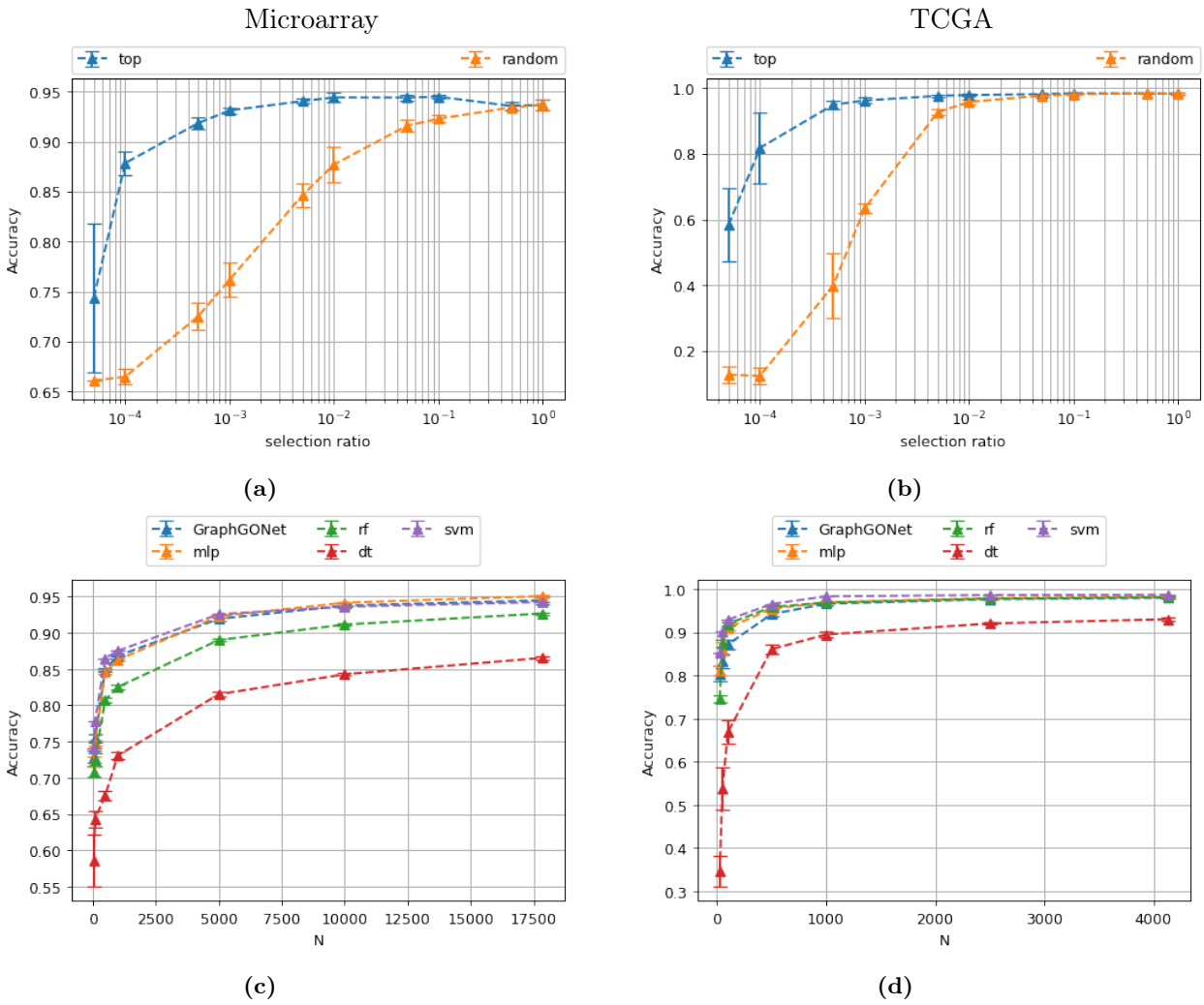


FIGURE 5.2 – ***Accuracy des modèles selon (a-b) le ratio de sélection*** r ***et (c-d) le nombre d'exemples d'apprentissage*** N ***sur les jeux de données microarray et TCGA.***

débutent autour d'une valeur moyenne de 0,733 à $r = 0,5$ pour atteindre à $r = 0,0005$ une accuracy de 0,57, correspondant à la proportion de la classe majoritaire. La sélection "top" permet d'inverser la tendance globale une fois de plus. En effet, les performances augmentent dans un premier temps de 0,754 à 0,780 avec des valeurs de sélection de 0,5 à 0,01, puis diminuent modérément de 0,780 à 0,683 avec un ratio compris entre 0,1 et 0,00005.

Les meilleures performances sont obtenues avec la sélection "top", avec un ratio de 0,1 sur les deux jeux de données publics et avec un ratio de 0,01 sur la cohorte. Il est intéressant de noter que les meilleures performances sont obtenues si la prédiction n'est basée que sur une petite proportion de neurones, d'environ 1000 à 100. Les termes GO associés devraient donc être liés au phénotype prédit. Cependant, des milliers de termes GO sont difficiles à assimiler par un utilisateur pour qu'il puisse en comprendre l'explication. Plus un sous-ensemble de termes GO sélectionnés est petit, plus l'explication sera compréhensible. Le ratio peut être vu comme un compromis entre performance et interprétation. Cet hyperparamètre est toujours ajustable en fonction des attentes de l'utilisateur.

Dans les expériences suivantes, nous considérons des modèles appris à partir de deux valeurs différentes de compromis qui sont les mêmes quel que soit le jeu de données. Le premier correspond au modèle le plus performant à $r = 0,01$ où une centaine de termes GO sont sélectionnés. Notons que ce point ne correspond pas précisément au ratio permettant d'obtenir les meilleures

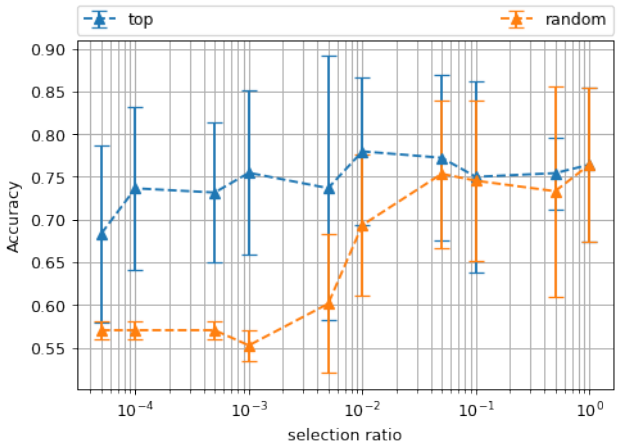


FIGURE 5.3 – ***Accuracy des modèles selon le ratio de sélection*** r ***sur le jeu de données d'Oncodesign.***

performances moyennes sur les deux jeux de données publics. Cependant, la différence de performances entre $r = 0,01$ et $r = 0,1$ pour les deux jeux de données est négligeable, puisqu'elle est inférieure à 0,005. Dans le second cas, nous faisons un compromis raisonnable entre les performances et l'interprétation en choisissant $r = 0,001$. L'accuracy diminue légèrement d'environ 2,5 % à 1,5 %, s'accompagnant d'une baisse du nombre de termes GO sélectionnés, passant d'une dizaine à une quinzaine en fonction du jeu de données.

Modèle	DT	RF	SVM	MLP	GraphGONet ($r = 0,01$)
Accuracy (moy.±std)	$0,62 \pm 0,09$	$0,72 \pm 0,08$	$0,72 \pm 0,07$	$0,77 \pm 0,10$	$0.78 \pm 0,09$

TABLE 5.2 – ***Comparaison des performances des modèles sur le jeu de données d'Oncodesign.***

Dans une deuxième expérience, nous comparons l'un des modèles proposés (à $r = 0,01$) avec des algorithmes ML classiques de l'état-de-l'art, comparables à ceux testés avec Deep GONet, soient l'arbre de décision abrégé DT (critère de Gini), RF (critère de Gini, nombre d'arbres=100), SVM (noyau linéaire, C=1,0), et MLP (trois couches avec respectivement 1000, 500 et 200 neurones). Sur les jeux de données publics, les méthodes sont entraînées sur différentes tailles d'ensembles d'entraînement, entre 17 847 (la taille complète de l'ensemble d'entraînement) à 50 échantillons pour le jeu de données microarray, et de 4136 à 25 échantillons pour le jeu de données TCGA. Comme pour l'expérience précédente, dix modèles ont été appris pour chaque taille d'échantillon. La Fig. 5.2c (resp., Fig. 5.2d) représente la moyenne et l'écart-type de l'accuracy de chaque méthode en fonction du nombre d'échantillons d'apprentissage du jeu de données microarray (resp., TCGA). Nous notons que les meilleures performances sont obtenues avec le plus grand nombre d'échantillons, et que les courbes des méthodes DL et SVM sont confondues pour les deux jeux de données. La forêt aléatoire est légèrement moins performante. Quant à l'arbre décision, la différence moyenne de performances avec les meilleures est supérieure à 10%. Les performances des modèles diminuent globalement avec un nombre réduit d'échantillons d'entraînement. En ce qui concerne la cohorte, nous n'avons pas fait varier la base d'apprentissage puisque la taille du jeu de données était déjà relativement petite. De même que pour GraphGONet, nous avons appris en validation croisée les méthodes ML classiques. Nous pouvons remarquer dans la Table 5.2 que contrairement aux deux autres jeux de données, les méthodes à base d'apprentissage profond et en particulier GraphGONet permettent d'obtenir les meilleures performances.

On a une différence de 5% avec la méthode SVM et RF et de 16 % avec l'arbre de décision. Ainsi, indépendamment de la taille de l'ensemble d'entraînement, GraphGONet est aussi compétitif que les algorithmes ML non explicables et surpasse clairement l'arbre de décision, la seule méthode interprétable par essence.

5.2.3 Analyse biologique

Cette sous-section montre comment proposer des interprétations biologiques pertinentes du modèle GraphGONet et de ses prédictions. Nous proposons deux niveaux d'interprétation : local et global.

a) Interprétation d'une prédiction d'un patient

Niveau GO	1	2	3	4	5	6	7	8	9
avg(nœuds_sélectionnés)	0,0866	0,0004	0,4912	0,4206	0,5086	0,6488	1,1705	2,0108	1,3591
std(nœuds_sélectionnés)	0,2813	0,0189	0,6048	0,5936	0,7366	0,7691	0,9959	1,1754	1,0145

Niveau GO	10	11	12	13	14	15	16	17	18	19
avg(nœuds_sélectionnés)	1,1909	0,8501	0,9545	1,0509	0,1667	0,0869	0,0022	0	0,0013	0
std(nœuds_sélectionnés)	0,8780	0,7893	0,8612	0,8502	0,4151	0,2818	0,0463	0	0,0354	0

(a)

Niveau GO	1	2	3	4	5	6	7	8	9
avg(nœuds_sélectionnés)	0,0650	0	0,3743	0,9002	0,3418	1,1833	1,3618	1,6357	1,9536
std(nœuds_sélectionnés)	0,2466	0	0,6050	0,8494	0,6494	1,1442	0,9170	1,1596	1,2175

Niveau GO	10	11	12	13	14	15	16	17	18	19
avg(nœuds_sélectionnés)	0,5275	0,1763	0,4749	1,4114	0,5947	0	0	0	0	0
std(nœuds_sélectionnés)	0,8421	0,3893	0,5011	0,7303	0,5926	0	0	0	0	0

(b)

Niveau GO	1	2	3	4	5	6	7	8	9	10
avg(nœuds_sélectionnés)	0	0,9565	0,0870	0,7391	0	1,1391	3,7391	0,9130	1,7391	0,5652
std(nœuds_sélectionnés)	0	0,2085	0,2881	0,6192	0	0,7168	1,2869	0,5964	1,0962	0,6624

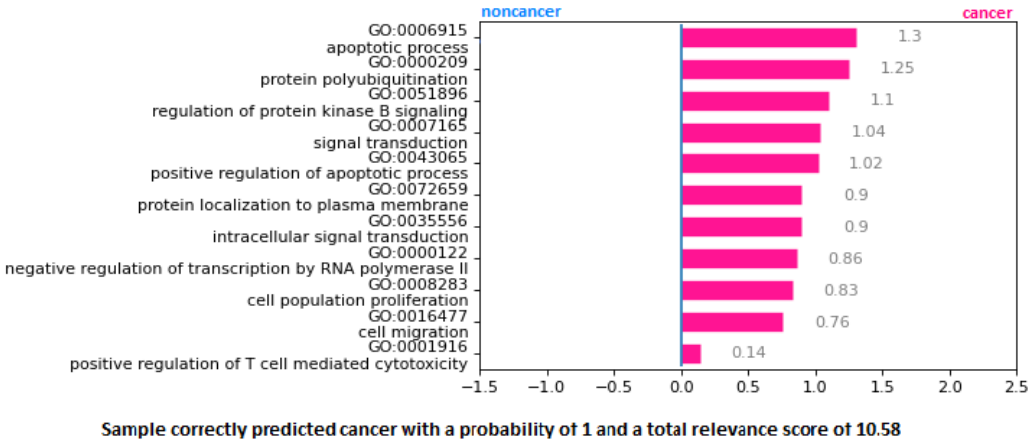
Niveau GO	11	12	13	14	15	16	17	18	19	20
avg(nœuds_sélectionnés)	1,7391	1,5652	2,3913	0,3478	0,0435	0	0	0	0	0
std(nœuds_sélectionnés)	1,0962	1,1211	0,8913	0,5728	0,2085	0	0	0	0	0

(c)

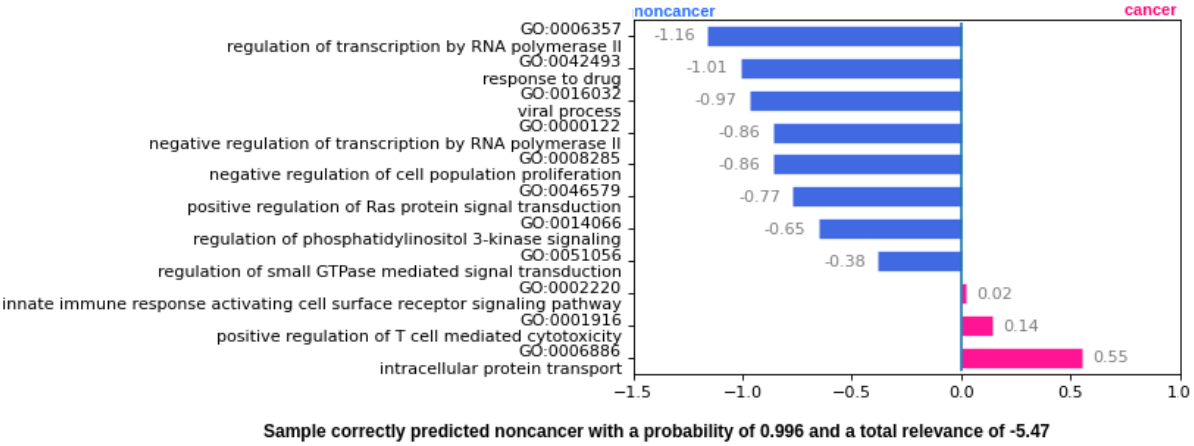
TABLE 5.3 – ***Moyenne et écart-type de la distribution des termes GO sélectionnés parmi les couches GO sur le jeu de données (a) microarray, (b) TCGA et (c) d'Oncodesign.***

Dans cette sous-section, nous montrerons comment fournir une explication de la classe prédite d'un patient calculée par GraphGONet. L'objectif est de proposer un outil prédictif et transparent aux utilisateurs finaux (biologistes, cliniciens, etc.), qui produit des explications claires et compréhensibles basées sur la connaissance en mettant en évidence l'ensemble de termes GO les plus impliqués dans la prise de décision avec leur contribution. Dans ce qui suit, nous utiliserons un ratio de sélection de $0,001$ qui conduit à un modèle utilisant seulement onze ou seize neurones et leurs termes GO associés pour réaliser les tâches de prédiction. Nous rappelons que la prédiction de chaque patient sera basée sur différents sous-ensembles de termes GO. La distribution moyenne des sous-ensembles parmi les couches GO est indiquée pour les trois jeux de données dans le tableau 5.3. Nous pouvons observer qu'en moyenne, les termes GO sélectionnés appartiennent à des niveaux intermédiaires entre sept et dix sur le jeu de données de microarray

et entre six et neuf sur le jeu de données TCGA et celui d'Oncodesign. Les scores de pertinence des termes GO sont comparés afin de détecter parmi le sous-ensemble les plus influents. Dans le cas d'une sortie softmax, plus le score est élevé, plus le terme GO a un impact positif sur la prédiction finale. Pour ce qui est de la sortie sigmoïde dans le cas des données microarray et de la cohorte, le signe de la contribution doit être interprété en tenant compte des résultats cancer ou non cancer (resp. répondeur ou non-répondeur). Les termes GO, qui contribuent à la prédiction cancer (resp. répondeur), ont un score de pertinence avec un signe positif. Si le signe est négatif, les termes GO ciblent la prédiction non-cancer (resp. non-répondeur).



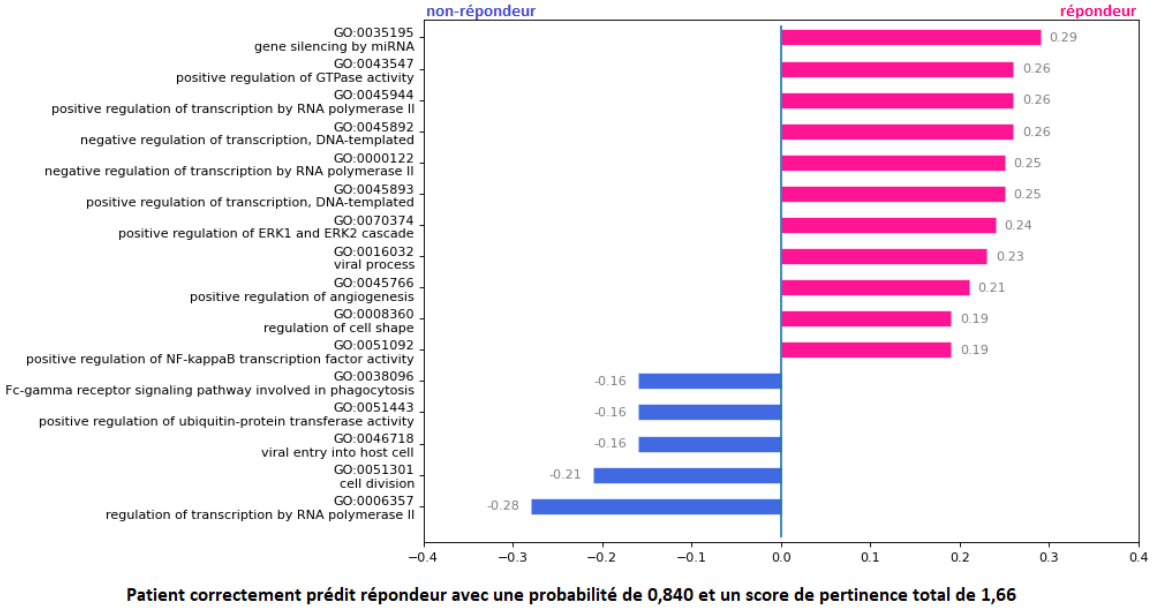
(a) *Patient n°6987*



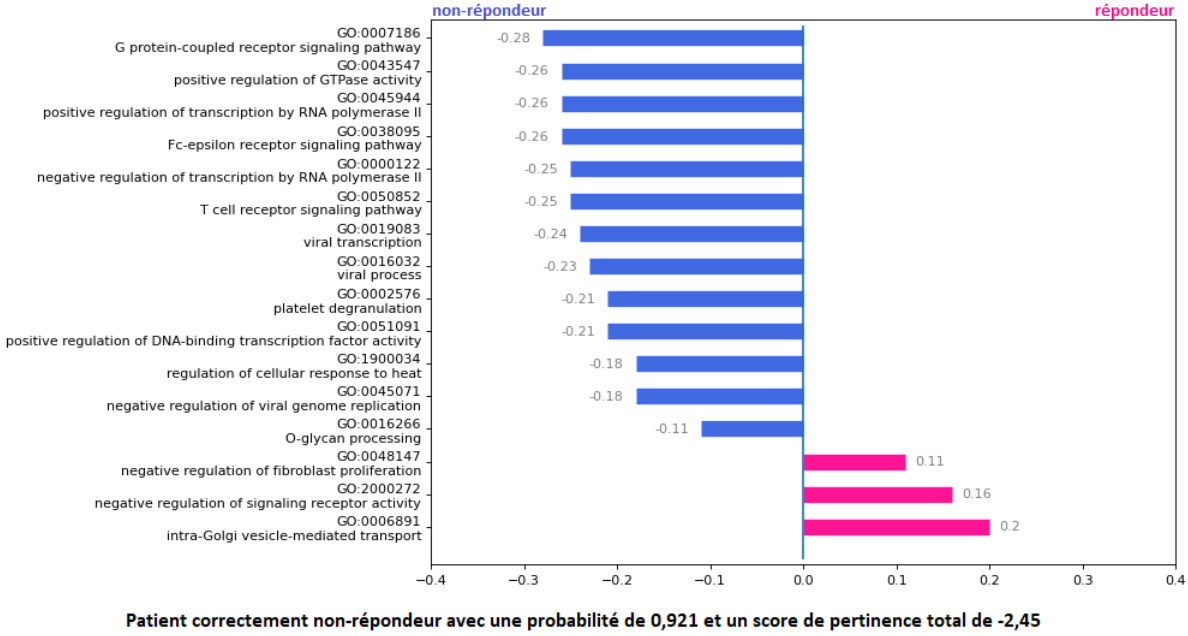
(b) *Patient n°2432*

FIGURE 5.4 – ***Explication (a) d'un patient prédit cancer et (b) d'un patient prédit non-cancer.*** *Les onze termes GO composant l'explication sont listés avec leur score de pertinence et leur description. La couleur indique vers quelle classe un terme GO influence le signal : bleu pour non-cancer et magenta pour cancer. Le score de pertinence total est la somme des scores de pertinence et du biais de la classe de sortie.*

Dans les Fig. 5.4a et b, nous illustrons l'application de notre outil sur un patient, à partir de l'ensemble de test de microarray, correctement prédit cancer avec une probabilité de 1 et un patient correctement prédit non-cancer avec une probabilité de 0,996. Les onze termes GO retenus sont rapportés avec leur score de pertinence selon un ordre descendant (ascendant) pour les patients cancer (non-cancer). Dans le cas des patients atteints de cancer, tous les termes GO ont un signe positif. Dix des onze termes GO sont importants pour la prédiction, car ils ont un score proche du score de pertinence moyen de 0,92. Seul le terme "GO :0001916" est moins important avec un score de 0,14. Parmi les termes GO les plus importants, nous pou-



(a) *Patiente répondeuse au traitement*



(b) *Patiente non-répondeuse au traitement*

FIGURE 5.5 – ***Explication (a) d'une patiente prédite répondeuse et (b) d'une patiente prédite non-répondeuse.*** *Les seize termes GO composant l'explication sont listés avec leur score de pertinence et leur description.*

vons en identifier certains qui pourraient jouer un rôle dans le cancer. Par exemple, les termes "GO :0006915" de rang un et "GO :0043065" de rang cinq sont liés à l'apoptose, ce qui peut mettre en évidence l'"immortalité" des cellules tumorales, c'est-à-dire leur capacité à se diviser à l'infini [LL00]. Nous pouvons observer que les termes "GO :0000122" et "GO :0001916" sont communs aux explications des patients cancer et non-cancer. Pour le terme "GO :0000122", la contribution est positive pour le patient cancer et négative pour le patient non-cancer, mais ils ont la même contribution absolue de 0,86 pour les deux prédictions. Le rang de ce terme GO est légèrement différent entre les deux exemples : sept dans la Fig. 5.4a et quatre dans la Fig. 5.4b,

mais il reste significatif pour les deux prédictions. Cela signifie qu'un terme GO peut être important pour les deux résultats. Le signal positif (négatif) détermine la prédiction vers le résultat de cancer (non-cancer). En revanche, le terme "GO :0001916" a un signe positif dans les deux cas. Il appartient à l'ensemble des trois termes de l'explication du patient non-cancer, qui ne codent pas pour la prédiction non-cancer. Par conséquent, le score de pertinence aide à repérer l'impact effectif des termes GO sur la prédiction finale et à quantifier l'incertitude de la prédiction. Des résultats similaires peuvent être obtenus sur la cohorte d'Oncodesign (voir Figures 5.5a-b). Sur les seize termes sélectionnés, plus du deux tiers ont un signe allant dans le sens de la prédiction répondeur/non-répondeur. Nous pouvons observer plusieurs termes en commun entre les explications des prédictions répondeur et non-répondeur : "GO :0043547", "GO :0045944", "GO :0000122" et "GO :0016032". Ces termes ont une contribution positive pour la patiente répondeuse et négative pour la patiente non-répondeuse. Leur rang est plus au moins identique entre les deux explications. Ils restent significatifs pour les deux prédictions. Une étude comparative des explications d'un patient prédisant un carcinome mammaire invasif (BRCA) (figure supplémentaire 7.5a) et un gliome cérébral de bas grade (LGG) (figure supplémentaire 7.5b) est fournie en annexe.

b) Interprétation du modèle

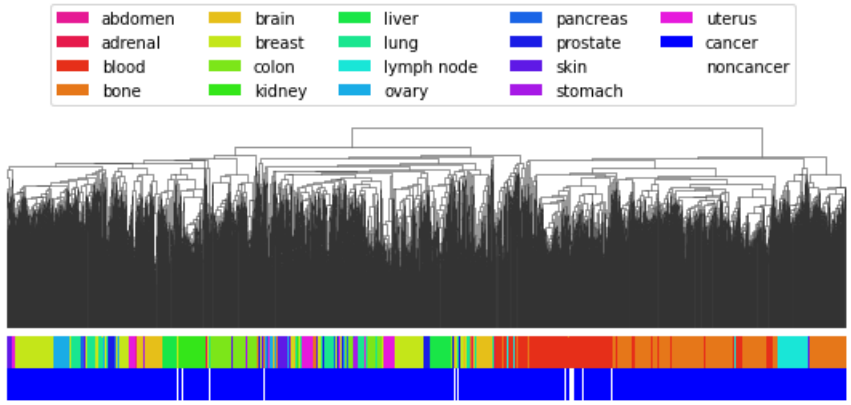


FIGURE 5.6 – ***Dendrogramme sur la matrice de pertinence des exemples cancer provenant du jeu de données microarray.*** *La première ligne indique le type de tissu de chaque exemple, tandis que la deuxième ligne indique la classe prédite.*

Type de tissu	abdomen	adrenal	blood	bone	brain	breast	colon	kidney	liver
#samples	10	15	454	623	168	366	218	98	120

Type de tissu	lung	lymph node	ovary	pancreas	prostate	skin	stomach	uterus	Total
#samples	169	112	95	36	77	79	27	116	2783

TABLE 5.4 – ***Liste des types de tissus observés dans les exemples cancer de l'ensemble de test microarray.*** *Le symbole # indique le nombre d'exemples.*

Dans cette sous-section, nous donnerons une interprétation globale du modèle, avec un ratio de sélection de 0,01, basée à la fois sur le score de pertinence et la fréquence des termes GO. Nous proposons d'abord d'analyser la similarité des explications entre les patients. Nous allons analyser le regroupement des échantillons de test prédits cancer en fonction de leurs profils de pertinence. Le profil de pertinence d'un patient est basé sur le score de pertinence de tous les neurones des couches GO. Les matrices de pertinence de taille (N, $|\mathcal{V}|$), où N est le nombre d'échantillons et $|\mathcal{V}|$

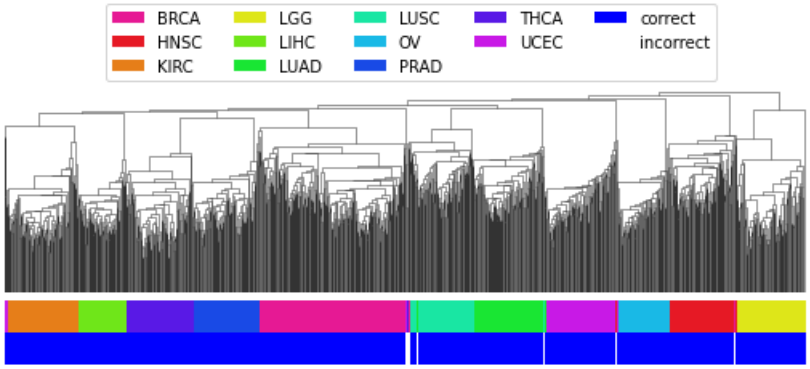


FIGURE 5.7 – ***Dendrogramme sur la matrice de pertinence des exemples cancer provenant du jeu de données TCGA.*** *La première ligne indique le type de cancer de chaque exemple et la deuxième ligne indique la justesse de la prédiction (blanc : mal classé, bleu : bien classé).*

le nombre de termes GO, sont collectées sur l'ensemble de test de chaque jeu de données. Une ligne correspond au profil de pertinence d'un patient. Nous appliquons la classification hiérarchique ascendante sur ces matrices, en utilisant la moyenne comme critère de liaison et la distance euclidienne comme métrique de distance. Le dendrogramme sur le jeu de données microarray est représenté sur la Fig. 5.6 et celui sur le jeu de données TCGA se trouve en Fig. 5.7. La première ligne colorée sous le dendrogramme indique le type de tissu de chaque échantillon (voir la Table 5.4 pour plus de détails). La deuxième ligne nous permet de distinguer la vraie prédiction (en bleu) de la prédiction incorrecte (en blanc). Nous pouvons discerner des clusters qui regroupent les patients provenant de mêmes tissus, en particulier pour le tissu osseux (coloré en orange), le tissu sanguin (coloré en rouge) et le tissu relatif au ganglion lymphatique (coloré en cyan). Bien que le modèle ne soit pas conçu pour prédire le type de tissu, certains neurones et leurs termes GO correspondants sont capables d'extraire des caractéristiques de cancer spécifiques à un type de tissu particulier. Une explication peut être que notre modèle n'identifie pas une signature unique du cancer, mais de multiples signatures associées aux différents tissus. Comme un modèle d'apprentissage profond produit une combinaison non-linéaire de plusieurs frontières de décision, GraphGONet réussit à identifier naturellement des signatures qui correspondent aux types de tissus. Nous pouvons enfin noter que les erreurs sont réparties entre les clusters. Cependant, une grande partie appartient au cluster lié au tissu sanguin. Il pourrait être intéressant d'étudier les raisons pour lesquelles ces exemples sont mal classés. Sur le jeu de données TCGA, chaque cluster formé correspond parfaitement à un type de cancer s'alignant sur l'objectif du modèle à prédire le type de cancer. Le modèle guidé par la connaissance biologique est capable de projeter les données dans un espace latent où les classes sont bien séparables.

Pour évaluer la cohérence des signatures biologiques, nous avons entraîné 100 modèles GraphGONet avec le même ratio de sélection de 0,01. Comme pour le résultat précédent, nous appliquons les modèles aux ensembles de test et calculons une matrice de pertinence et une matrice d'occurrences. La dimension de ces matrices est $(S, N, |\mathcal{V}|)$, où S correspond au nombre de modèles. La matrice d'occurrences est une matrice booléenne indiquant si un terme GO a été sélectionné ou non par la couche de sélection. Nous pouvons alors sommer sur l'axe du modèle et l'axe du patient le nombre de fois qu'un terme GO est sélectionné, ce qui donne un vecteur de taille $|\mathcal{V}|$. Les Fig. 5.8a-c montrent, respectivement, que 40,34 % des termes GO pour microarray, 62,79 % pour TCGA et 59,73 % pour la cohorte d'Oncodesign n'ont jamais été conservés par la phase de sélection. À l'inverse, certains termes GO sont souvent sélectionnés par le module de sélection de la plupart des modèles. Ils doivent contenir des informations biologiques pertinentes pour les prédictions. C'est le cas du terme "GO :0045944" le plus fréquent, qui apparaît 403K fois dans les expériences sur microarray, 88,3K fois sur TCGA et 1,2K fois sur la cohorte

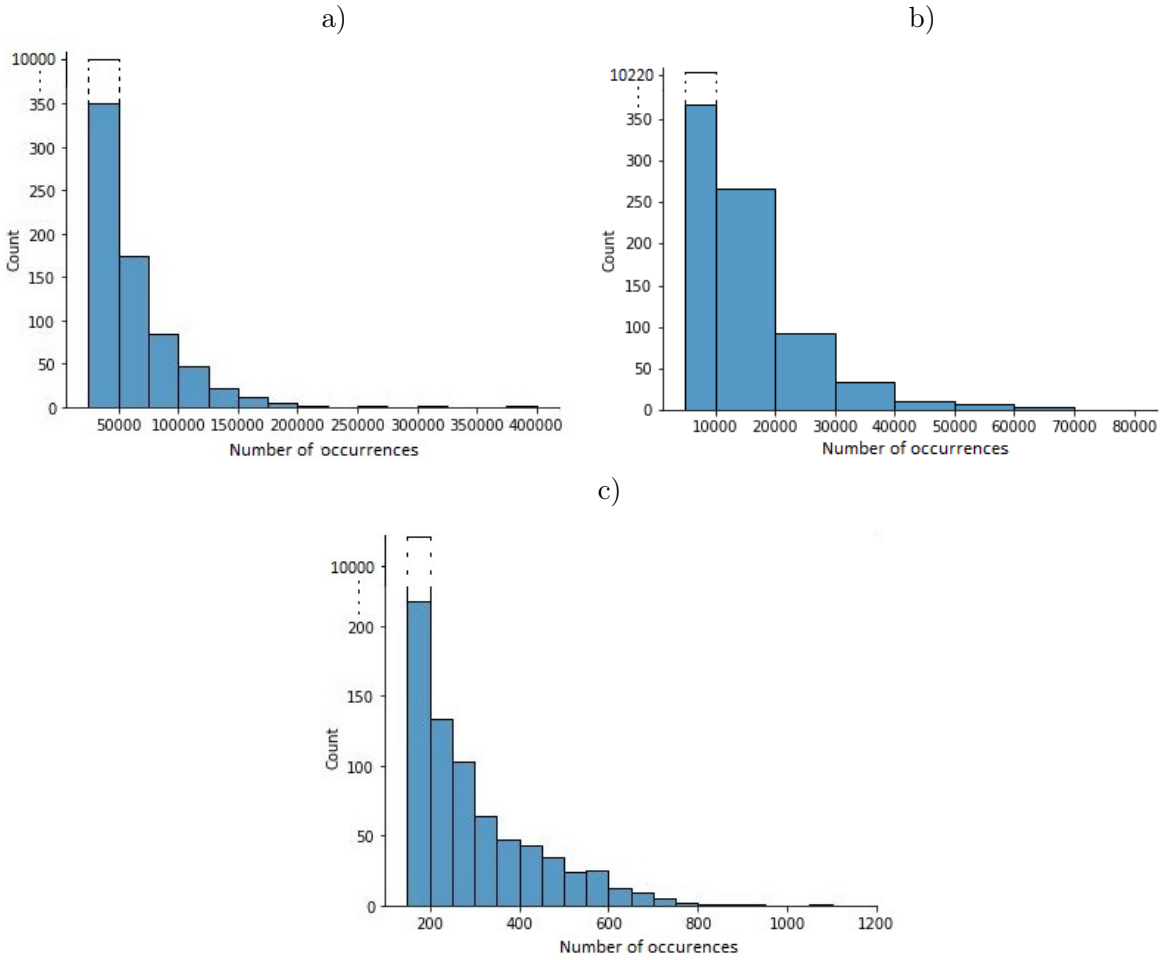


FIGURE 5.8 – ***Histogramme des occurrences basé sur la matrice d'occurrences sur (a) microarray, (b) TCGA et (c) la cohorte.*** *L'axe des abscisses correspond au nombre de fois où un terme GO est sélectionné dans l'ensemble des modèles et des exemples et l'axe des ordonnées est le nombre de termes GO prenant valeur dans les intervalles. La fréquence maximale qui peut être atteinte correspond au nombre d'exemples testés multiplié par le nombre de modèles, soit 557,8K sur microarray, 129,3K sur TCGA et 2,3K sur la cohorte.*

(respectivement, 72 %, 68 % et 51 % de l'occurrence maximale qui peut être atteinte).

Pour obtenir une interprétation du modèle en tenant compte des classes, nous filtrons ces matrices en fonction de l'étiquette de chaque patient et classons les termes GO en fonction de leur occurrence. Nous comptabilisons le nombre de signes positifs ou négatifs des contributions des termes GO sur les deux premières dimensions. Cela indique si le terme GO est utilisé en faveur de la prédiction dans le cas d'un signe positif ou contre la prédiction dans le cas contraire. La Fig. 5.9 montre un exemple des dix termes GO les plus fréquents sur le jeu de données TCGA pour le cancer BRCA. Nous constatons que ces termes GO ont une occurrence proche de la limite supérieure de 22 100 (cas où ils sont sélectionnés pour chaque patient par tous les modèles). Par exemple, le terme "GO :0000122" classé en première position ressort 15 307 fois. De plus, cet histogramme met en évidence que dans la plupart des cas, plus un terme GO est fréquent, plus ce terme GO est utilisé pour prédire l'étiquette de la classe. Nous pouvons identifier des cas particuliers grâce aux termes "GO :0006614" et "GO :0006958". Ils sont utilisés dans les deux sens, mais la moyenne de leur pertinence absolue révèle qu'ils ne contribuent pas autant dans le calcul de la prédiction, contrairement aux autres termes GO les plus importants. Des résultats comparables peuvent être générés pour chaque type de cancer. Nous observons

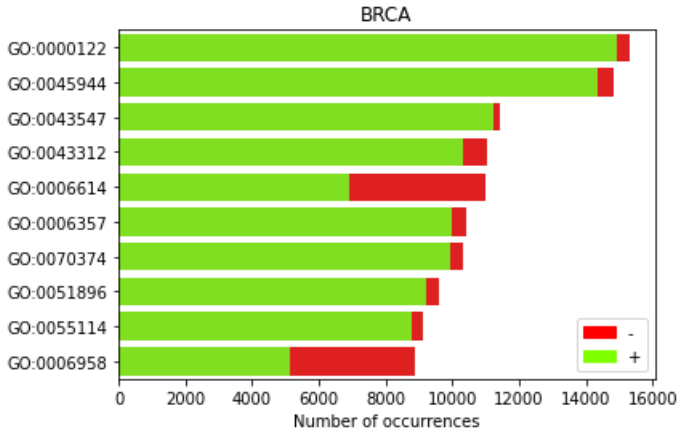


FIGURE 5.9 – *Top-10 des termes GO les plus fréquents triés en fonction de leur occurrence pour le type de cancer BRCA à partir de TCGA.* *Les couleurs indiquent la proportion des occurrences ayant un score de pertinence négatif (rouge) ou positif (vert). L'occurence maximale pouvant être atteinte correspond au nombre d'exemples BRCA multiplié par le nombre de modèles, soit 22 100.*

que certains termes GO sont importants pour tous les types de cancer, comme "GO :0045944", alors que d'autres sont spécifiques à certains types de cancer. Sur les jeux de données microarray et d'Oncodesign, les résultats des figures supplémentaires 7.6 en annexe montrent à nouveau que les termes GO peuvent être utilisés pour les deux prédictions, mais le score de pertinence sera positif pour la prédiction cancer (resp. répondeur) et négatif pour celle non-cancer (resp. non-répondeur).

5.3 Discussion et conclusion

Nos expériences ont montré que GraphGONet peut tirer parti de l'ensemble du graphe de connaissances et de sa sémantique pour réaliser la tâche de prédiction de manière efficace. Le modèle a été validé sur trois jeux de données et peut traiter toute connaissance de la forme d'un DAG. Nous avons apporté une nouveauté dans la diffusion de l'information au travers de la combinaison d'un FFNN et d'un GNN et par la couche de sélection, qui permettent de combiner l'expression de gènes dans l'apprentissage et de fournir automatiquement des informations biologiques pour la prise de décision.

Plus précisément, la propagation dans les couches GO est issue du processus de propagation utilisé dans les couches de convolution des GNNs. Cela permet d'inclure tous les niveaux de GO et tout type de relation (adjacente, non adjacente) entre les termes GO. Le processus de propagation diffère légèrement du modèle GCN [KW17] du fait qu'on dispose de deux paramètres, l'un pondérant l'information provenant du voisinage et l'autre modulant l'information provenant des gènes qui pourrait être vue comme la représentation vectorielle initiale des nœuds. Nous aurions pu pour un neurone GO donné moyenner l'information provenant des gènes associés au lieu de passer par une combinaison linéaire, mais il nous a paru plus pertinent de laisser le modèle combiner automatiquement cette information.

Ensuite, une différence majeure par rapport à la plupart des méthodes GNNs de l'état de l'art est que majoritairement ce sont des graphes non orientés qui sont traités et les couches de convolution parallélisent généralement la mise à jour des représentations vectorielles des nœuds. Dans notre approche, la mise à jour est séquentielle de sorte à pouvoir propager l'information le long de la hiérarchie de connaissances et de toujours prendre l'information d'un nœud fils la plus à jour possible. Ce principe a été également proposé par [TC21] pour traiter les graphes orientés et en l'occurrence les DAGs. Ce processus a permis dans notre cas de ne pas avoir à appliquer plusieurs couches de convolution pour atteindre de bonnes performances. Une autre distinction

importante est qu'autre l'orientation des arêtes, les graphes pour des tâches de classification ont généralement une petite taille (pas plus de 500 nœuds) alors qu'on est autour de 10,5K nœuds. De plus, tous les patients disposent du même graphe GO, seul le signal diffère. La sélection reste personnalisable à chaque patient.

Concernant cette couche de sélection employée pour réduire le graphe, elle se situe au croisement de la méthode SortPool et des méthodes top-k. Du fait d'une seule couche de convolution, de la topologie hiérarchique de notre graphe et de l'information portée par un nœud réduite à un scalaire réel, passer par une projection ou par l'attention pour déterminer le score de chaque nœud n'apporterait rien. Généralement, ces techniques sont utilisées lorsque l'information portée par un nœud est à valeur dans un espace multidimensionnel. Similairement à SortPool, nous avons basé le score d'importance d'un nœud selon sa représentation scalaire, soit la valeur d'activation du neurone. La méthode de clustering proposée par DiffPool, quant à elle, a été développée pour des graphes non orientés et semblait donc difficilement applicable à un DAG. Les *supra*-nœuds créés paraissaient également peu interprétables. Dans DAGNN, seul un module de compression *readout* a été utilisé où uniquement l'information contenue par les nœuds racines du DAG est gardée à chaque couche de convolution via des connexions résiduelles. Une opération d'agrégation est ensuite appliquée sur la concaténation vectorielle obtenue pour retenir le maximum d'information portée par dimension. En effectuant cette opération, nous perdons toutes les informations sur les termes GO choisis. De plus, comme il ne s'agit pas du même ensemble de nœuds sélectionnés d'un patient à l'autre, nous avons opté pour une opération de masquage au lieu d'une concaténation, afin de préserver la notion de concept biologique, propre à notre problème. Ceci permet aussi d'avoir accès directement à l'ensemble final de neurones, et leurs fonctions biologiques associées, utilisés pour le calcul de la prédiction finale et servant de support de l'explication rendue automatiquement.

Ces différentes caractéristiques ont permis de rendre GraphGONet une méthode *self-explaining* au même titre que les méthodes à base de concepts telles que *Concept Bottleneck* [Koh+20a], contrairement aux GNNs cités qui nécessitent généralement l'usage d'une méthode a posteriori adaptée pour les démystifier [Yua+20]. Comme évoqué dans l'état de l'art, les quelques travaux sur les GNNs pour les données moléculaires appliquent des GNNs non *self-explaining* comme GCN sur des graphes IPP où une protéine est associée à son gène correspondant [RSK18 ; Ram+20 ; Che+21]. À notre connaissance, aucune approche n'utilise l'ontologie GO à ce jour.

Nous allons maintenant comparer notre nouvelle approche à la précédente. GraphGONet offre plusieurs avantages par rapport à Deep GONet. Tout d'abord, le sous-ensemble des termes GO composant l'explication d'une prédiction est individualisé pour chaque patient. La couche de sélection dans GraphGONet le permet en choisissant localement les neurones les plus informatifs et leurs termes GO associés. Les termes peuvent appartenir à différents niveaux de l'ontologie d'origine. Au contraire, dans Deep GONet, l'explication est moins personnalisée. Le terme L_{GO} régularisant les poids du modèle, en l'occurrence ceux des connexions noGO, a une portée globale. De plus, la méthode ne rend pas automatiquement un sous-ensemble de termes les plus importants. Le recours à une méthode de décomposition a été nécessaire pour identifier par couche les termes formant ce sous-ensemble. Contrairement à GraphGONet où l'hyperparamètre r a permis d'ajuster raisonnablement la taille de ce sous-ensemble pour favoriser sa compréhension, il est moins évident de définir ce seuil dans Deep GONet. Cela dépendra de la distribution des scores de pertinence d'un patient. Dans les exemples présentés, il avait été fixé à cinq, pour un total de trente termes, toutes couches confondues. Notons que ce seuil n'aura pas d'influence sur l'accuracy contrairement à l'hyperparamètre r puisqu'il intervient seulement dans l'analyse a posteriori. Certes, une analyse supplémentaire a été réalisée sur le sous-ensemble fourni par GraphGONet pour déterminer la contribution relative de chaque terme de celui-ci, mais la méthode a posteriori utilisée ne nécessite en réalité aucun calcul supplémentaire. Les scores peuvent être directement extraits à partir de l'inférence. Nous avons également montré que le compromis entre

interprétation et performance est moindre par rapport à Deep GONet. En passant de $r = 0.01$ à $r = 0.001$, nous avons perdu 1.5 % d'accuracy alors que dans Deep GONet, on a perdu environ 3 % en passant de $\alpha = 10^{-5}$ à $\alpha = 10^{-2}$.

Ensuite, dans GraphGONet, nous avons pu prendre en compte plus de sémantique, notamment les relations entre couches non adjacentes qui n'avaient pas été considérées dans Deep GONet. Dans GraphGONet, les gènes peuvent être connectés à toutes les couches GO et les neurones de couches GO non adjacentes peuvent être connectés entre eux. Dans le cadre d'un MLP tel que Deep GONet, chaque connexion est pondérée. Ainsi, plus les connexions sont nombreuses, plus le modèle doit apprendre de paramètres. On est à plus de 4,1M de paramètres seulement entre les couches cachées de Deep GONet contre deux paramètres partagés par l'ensemble des couches cachées dans GraphGONet. Pour la même connectivité que ce dernier, nous aurions eu environ 23 900 paramètres pour un MLP avec des connexions résiduelles. Même si ce nombre est réduit par rapport à un réseau de neurones totalement connecté, un GNN offre plus de flexibilité qu'un MLP avec des connexions résiduelles. En effet, on peut ajouter autant de connexions entre les neurones des couches GO, il n'y aura pas plus de paramètres à apprendre. Le nombre total de paramètres dans GraphGONet inclut les poids des connexions entre les gènes et les neurones GO et ceux entre la couche de sélection et la couche de sortie. Sur le jeu de données microarray, on est à 291 859 paramètres.

Le seul désavantage de GraphGONet par rapport à Deep GONet est que les gènes non annotés n'ont pas été inclus, néanmoins on gagne en consistance. Une solution qui pourrait être envisagée serait de connecter ces gènes aux mêmes termes GO des gènes annotés avec qui ils sont le plus corrélés. Nous pourrions ensuite appliquer une pénalité pour distinguer les deux types de liens.

L'utilisation de GraphGONet présente d'autres avantages. Les réseaux de neurones actuels utilisent généralement des méthodes a posteriori pour estimer la pertinence de chaque gène, ce qui donne lieu à un large ensemble de scores de pertinence rendant les explications moins compréhensibles. En revanche, GraphGONet fournit une explication accessible et compréhensible des prédictions aux experts en biologie en produisant un petit ensemble de termes GO, avec leurs scores de pertinence associés, sur une base d'une connaissance avec laquelle les utilisateurs finaux sont familiers. L'un des avantages de GraphGONet est que l'explication est basée sur un sous-ensemble de concepts de haut niveau sémantique - des termes GO au lieu de gènes - ce qui a permis d'obtenir des interprétations plus stables. Tout d'abord, nous considérons par consistance le fait que l'apprentissage de plusieurs modèles ayant exactement la même configuration, mais où les paramètres sont initialisés de façon différente, devrait aboutir à des explications équivalentes. L'utilisation de méthodes a posteriori montre que dans certains cas, un modèle non *self-explaining* peut donner lieu à des explications non équivalentes. Les réseaux de neurones formés sur le même ensemble d'apprentissage peuvent conduire à des explications différentes en raison des différentes initialisations des paramètres du modèle [Elt20]. Nous avons montré que les explications biologiques produites par notre modèle sont stables (cf. : Fig. 5.8 et Fig. 5.9), puisque les ensembles de termes GO retournés étaient similaires dans les cent modèles, chacun ayant été appris avec une initialisation différente de ses paramètres. Plusieurs études ont montré que l'identification des signatures génétiques obtenues à partir de méthodes d'interprétation de modèles ou de sélection de caractéristiques est très instable [DHZ12]. Bien que l'ensemble des gènes les plus importants pour la prédiction puisse toujours être identifié par des méthodes post hoc, les signatures basées sur les termes GO semblent plus fiables.

Un autre avantage est que le score de pertinence de chaque terme GO est facile à calculer et représente un bon indicateur qui quantifie la contribution finale des termes dans le sous-ensemble. En pratique, ces scores dépendent simplement des poids de leurs neurones associés à la couche de sortie, et du signe de l'activation des neurones. Cette facilité vient du fait qu'un terme GO, représenté par un neurone, est directement connecté à la couche de sortie. Nous postulons que le

signe du score de pertinence est essentiel pour indiquer de quelle manière les termes GO les plus importants influencent le résultat vers une prédiction vraie ou fausse. Un changement dans le signe du signal peut faire passer le résultat prédit d'une classe à une autre. L'examen des raisons de ce changement et la compréhension de la signification du signe (par exemple, si le neurone représentant le terme GO est excité ou inhibé) nécessitent l'aide d'un expert biologique.

Un troisième avantage est que différents niveaux d'interprétation peuvent être proposés en fonction des attentes de l'utilisateur final. Comme nous l'avons vu dans l'étude de l'ablation sur le ratio de sélection, l'accent mis sur l'interprétation plutôt que sur la performance dépend des exigences de l'utilisateur final. Une explication produite avec cent termes GO peut encore être comprise par des experts en biologie. Il est possible de recréer le sous-graphe associé à un patient et d'en identifier les parties les plus pertinentes. Selon l'utilisateur final, différents types d'explication peuvent être fournis : un médecin ou un patient peut être plus intéressé par l'interprétation de la prédiction d'un patient, alors qu'un biologiste et un statisticien par l'interprétation globale du modèle.

Un dernier avantage est que la fiabilité du système peut être estimée à partir de la performance et de l'interprétation offerte par le modèle. Nous considérons qu'un modèle est fiable en fonction des performances mesurées et de la pertinence des explications. La performance est un critère qui peut être quantifié par des benchmarks. Dans notre analyse de sensibilité, nous avons montré à travers les métriques de précision, AUPRC et le F1-Score que GraphGONet est à la fois explicable et aussi compétitif que les autres méthodes ML. Bien qu'il n'existe pas de métrique standard pour mesurer la pertinence des explications, nous pouvons vérifier si le sous-ensemble de termes GO est cohérent avec le phénotype donné. Une recherche manuelle dans la littérature devrait alors être effectuée pour quantifier les liens entre les termes GO retournés et le phénotype prédit, comme cela fut présenté dans l'analyse de l'explication de prédiction de la Fig. 5.4. La pertinence et la causabilité (*causability* en anglais), ou la qualité des explications [HCM20], peuvent donc être évaluées par des experts en biologie sur la base du sous-ensemble de termes GO, de leur score de pertinence et d'une validation bibliographique.

Plusieurs pistes potentielles de recherche peuvent être envisagées pour étendre les capacités de GraphGONet. D'une part, bien que les étiquettes sur les arêtes n'aient pas été incluses, nous pourrions modifier légèrement la loi de propagation des couches convolutives pour les prendre en compte. D'autre part, il est également possible d'inclure plusieurs connaissances. Une extension directe de GraphGONet serait de par exemple connecter la couche de gènes à une nouvelle branche représentant une ontologie de type DAG comme Reactome pour continuer à utiliser la loi de propagation proposée (Eq. (5.1)) et d'étendre la couche de sélection d'un nombre de neurones complémentaires représentant les concepts biologiques de la connaissance additionnelle. De cette sorte, nous pourrions continuer à obtenir des explications automatiques basées sur les concepts biologiques les plus importants. Une alternative serait de transformer la couche de gènes en un graphe non orienté basé, par exemple sur le graphe de connaissances IPP, et le connecter à diverses ontologies qui peuvent ne pas être des DAGs. Les GNNs permettent de propager l'information dans des graphes de type et taille divers. Cela aurait l'avantage d'enrichir les explications et d'identifier les gènes importants automatiquement, tout en maintenant la consistance des signatures grâce au guidage par la connaissance.

Chap.

6 BioHAN

Contenu du chapitre

Par cette dernière contribution, nous poursuivons notre travail entrepris avec GraphGONet sur les GNNs comme modèles *self-explaining* et interprétables par construction. Nous proposons notamment d'évaluer la flexibilité offerte par ce type de réseau de neurones en les enrichissant d'autres connaissances biologiques. Le nouveau modèle GNN ainsi créé est construit par blocs de convolution-pooling sur des graphes hétérogènes. De plus, l'entité "gène" n'est plus simplement un neurone d'entrée, comme précédemment, mais un nœud central de l'approche GNN proposée afin d'y avoir accès directement pour l'interprétation. Les connaissances support telles que GO restent exploitées pour permettre de faire communiquer dans le GNN l'ensemble des gènes à distance variable, et compléter et maintenir l'intelligibilité des explications biologiques. Le recours à la technique de l'attention permet enfin aux nœuds de se concentrer sur les voisins les plus informatifs et facilite, sans surcoût calculatoire, l'analyse a posteriori retraçant le parcours du signal jusqu'aux gènes les plus pertinents. Dans un premier temps, nous présenterons de manière détaillée ce nouveau modèle, intitulé BioHAN pour *Biological Heterogeneous Attention Network*. Puis, nous présenterons les premiers résultats sur un jeu de données réelles ainsi qu'un exemple d'interprétation possible avec cette nouvelle méthode. Enfin, nous discuterions des contributions et des améliorations envisagées de ce troisième travail.

6.1 Méthode

BioHAN est exclusivement basée sur un GNN intégrant les quatre modules de base pour la classification de graphes, à savoir la couche de convolution, les couches de réduction de *pooling* et de *readout*, et enfin la couche dense. L'approche complète est illustrée par la Fig. 6.1. Chaque couche est décrite une par une dans la sous-section suivante.

6.1.1 Description de l'architecture

Comme précédemment introduit, l'information est centralisée autour d'un sous-graphe central parcimonieux dont les nœuds représentent des gènes. Nous avons toujours recours à la connaissance issue des ontologies pour aider à l'interprétation des résultats, mais également à la propagation du signal dans le sous-graphe central malgré sa parcimonie. La parcimonie du sous-graphe

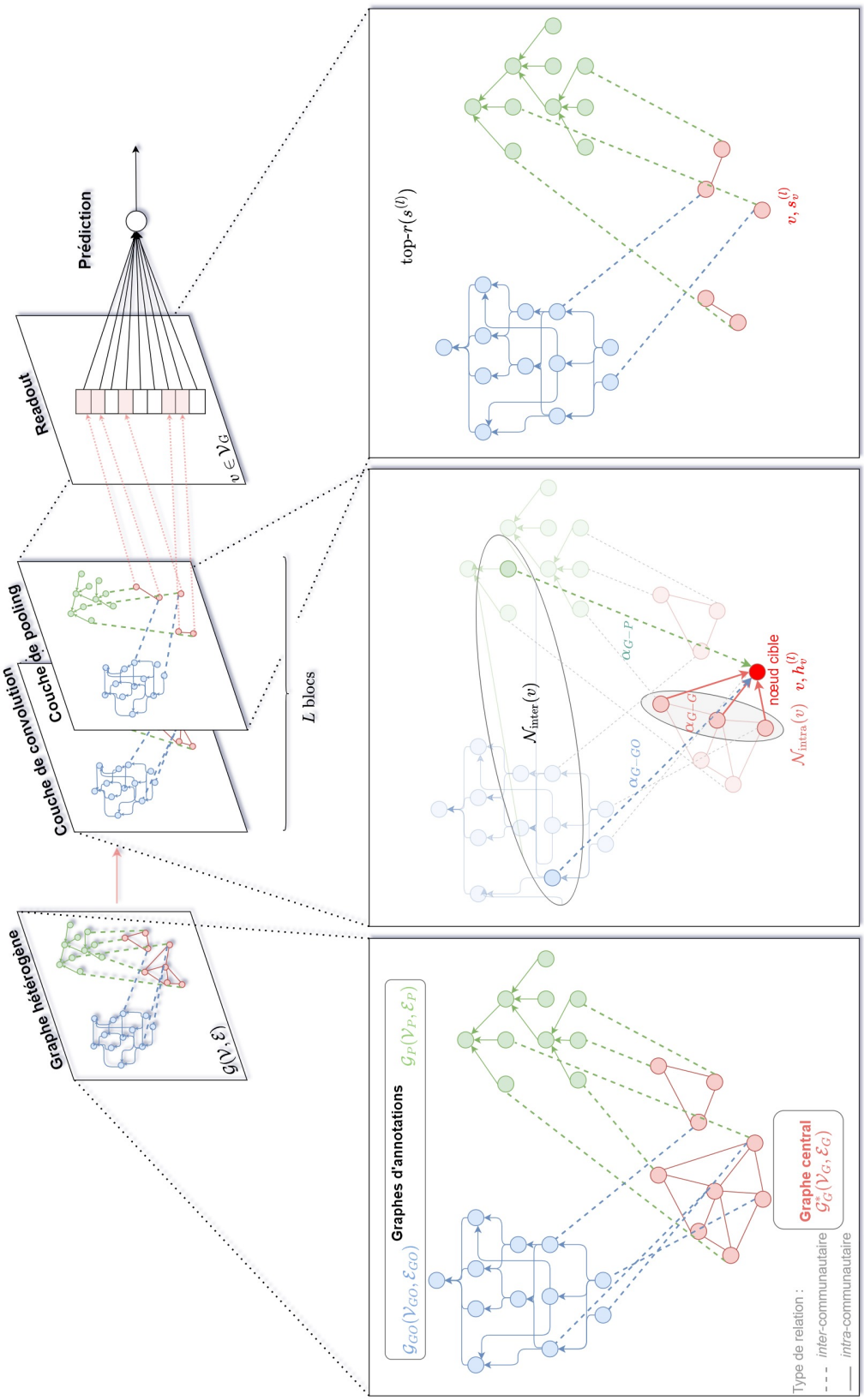


FIGURE 6.1 – ***Illustration de l'approche BioHAN.*** *Le graphe hétérogène d'un patient est ici formé d'un sous-graphe central de gènes $\mathcal{G}_G^*$ en rouge basé sur le sous-graphe d'interactions protéines-protéines (IPP) et de deux sous-graphes auxiliaires : GO ($\mathcal{G}_{GO}$) en bleu et Reactome ($\mathcal{G}_P$) en vert. Les étapes de convolution et de pooling sont illustrées par les encadrés en noir.*

central aura de plus tendance à augmenter à cause des couches de réduction. La connaissance auxiliaire permettra alors de préserver la connectivité du sous-graphe central grâce à des nœuds associés à des termes d'annotation et reliés à des gènes potentiellement distants, voire isolés. Ces nœuds permettront de faire communiquer ces gènes indirectement.

Chaque patient (X, Y) est ainsi représenté par un graphe hétérogène de connaissances $\mathcal{G}(\mathcal{E}, \mathcal{V})$ qui regroupe un ensemble I de sous-graphes d'annotations, notés $\mathcal{G}_{\mathcal{K}_i}$, centrés autour d'un sous-graphe central de gènes, noté $\mathcal{G}^*_G$, qui sera connecté à tous les sous-graphes $\mathcal{G}_{\mathcal{K}_i}$. Nous avons $\mathcal{G} = \{\mathcal{G}^*_G, \mathcal{G}_{\mathcal{K}_1}, \mathcal{G}_{\mathcal{K}_2}, \ldots, \mathcal{G}_{\mathcal{K}_I}\}$ ($\mathcal{E} = \{\mathcal{E}_G, \mathcal{E}_{\mathcal{K}_1}, \mathcal{E}_{\mathcal{K}_2}, \ldots, \mathcal{E}_{\mathcal{K}_I}\}$, $\mathcal{V} = \{\mathcal{V}_G, \mathcal{V}_{\mathcal{K}_1}, \mathcal{V}_{\mathcal{K}_2}, \ldots, \mathcal{V}_{\mathcal{K}_I}\}$) où chaque sous-graphe de connaissances $\mathcal{G}_j$ se définit par un ensemble $\mathcal{V}_j$ de nœuds de même type et un ensemble $\mathcal{E}_j$ d'arêtes qui se différencient en *intra* et *inter*-communautaires. Les arêtes *intra*-communautaires, pouvant être orientées ou non orientées, connectent deux nœuds de même type provenant du même sous-graphe de connaissances, tandis que les arêtes *inter*-communautaires relient deux nœuds de type différent appartenant à des sous-graphes de connaissances distincts. On définit par voisinage d'un nœud v l'ensemble des nœuds d'ordre 1 qui lui sont connectés directement. Le nœud v peut être lui-même inclus dans son voisinage. On discerne alors plusieurs voisinages d'un nœud v en fonction des nœuds avec lesquels il interagit. En fonction de sa connectivité avec les autres sous-graphes de connaissances, il peut avoir B ($1 \leq B \leq I$) voisinages *inter*-communautaires qu'on désignera par $\mathcal{N}^j_{inter}(v)$ où $j = 1, \ldots, B$. Cependant, un nœud n'aura toujours qu'un seul voisinage *intra*-communautaire vis-à-vis de son sous-graphe d'appartenance, noté $\mathcal{N}_{intra}(v)$. Au total, un nœud peut avoir autant de voisinages qu'il y a de sous-graphes de connaissances. L'information contenue dans chacun des nœuds est fournie par un vecteur de dimension D, pouvant varier en fonction du type de nœud. Dans le cas de l'application aux profils d'expression, $D = 1$ quel que soit le type de nœud. La propagation de cette information $h_v \in \mathbb{R}^D | v \in \mathcal{V}$ se fait par l'intermédiaire des couches de convolution de graphe. La couche de convolution choisie est basée sur un mécanisme d'auto-attention pour permettre aux nœuds de ne se focaliser que sur certains voisins [Vel+18 ; Wan+20]. L'attention favorisera également l'interprétation des prédictions. Elle sera dépendante du type de nœud et du type de relation entre deux nœuds. Les représentations vectorielles portées par les différents types de nœuds sont préalablement projetées dans un espace commun par les poids $w_{t(v)} \in \mathbb{R}^{D' \times D}$ où la fonction $t(v)$ associe à un nœud v un type dans l'espace de types $\mathcal{T}$ tel que $\mathcal{T} = \{G, GO, P\}$ dans la Fig. 6.1. Les nœuds issus du même sous-graphe de connaissances partagent le même paramètre w. Un score d'alignement noté e_{vu} est ensuite calculé entre chaque paire de nœuds v et u, que l'on formalise par la fonction d'attention $a : \mathbb{R}^{D'} \times \mathbb{R}^{D'} \to \mathbb{R}$ telle que $e_{vu} = a(w_{t(v)}h_v, w_{t(u)}h_u)$. Cette fonction correspond à un perceptron et est paramétrisée par le poids d'attention $w_{\alpha, r(v,u)} \in \mathbb{R}^{2D'}$ telle que :

$$e_{vu} = \text{LeakyReLU}\Big(w^T_{\alpha, r(v,u)}\big[w_{t(v)}h_v \| w_{t(u)}h_u\big]\Big) \tag{6.1}$$

où $\|$ désigne l'opérateur de concaténation, $.^T$ la transposée et la fonction $r(v, u)$ la relation entre deux nœuds dans l'espace de relations $\mathcal{R}$ tel que $\mathcal{R} = \{G-G, G-GO, GO-GO \ldots\}$. La fonction d'activation LeakyReLU correspond à une variante de la fonction ReLU avec une faible pente pour les valeurs négatives au lieu d'une pente plate telle que $\text{LeakyReLU} = \max(0, x) + ns \times \min(0, x)$ où ns désigne la pente négative (*negative slope* en anglais).

La mise à jour d'un nœud $v \in \mathcal{V}$ par une couche de convolution l s'exprime finalement de la manière suivante :

$$\tilde{h}^{(l)}_v = GCONV(h^{(l-1)}_v) \tag{6.2}$$

$$= \sigma\left(\frac{1}{(B+1)K}\sum_k^K\Big(\sum_{u \in \mathcal{N}^{(l)}_{intra}(v)} \alpha^{(l,k)}_{uv} w^{(l,k)}_{t(u)} h^{(l-1)}_u + \sum_j^B \sum_{u \in \mathcal{N}^{(j,l)}_{inter}(v)} \alpha^{(l,k)}_{uv} w^{(l,k)}_{t(u)} h^{(l-1)}_u\Big)\right) \tag{6.3}$$

où $h^{(l-1)}_u$ représente la représentation vectorielle du nœud u à la couche précédente $(l-1)$, $\alpha^{(l,k)}_{vu}$

correspond à un coefficient d'attention normalisé tel que $\alpha_{vu}^{(l,k)} = \text{softmax}(e_{vu}^{(l,k)}) = \frac{\exp(e_{vu}^{(l,k)}/\tau)}{\sum_{p\in\mathcal{N}(v)}\exp(e_{vp}^{(l,k)}/\tau)}$ avec τ le coefficient de température permettant d'ajuster la parcimonie du coefficient d'attention, $\mathcal{N}_{(\cdot)}^{(\cdot,l)}(v)$ désigne un voisinage de v (inter ou intra), σ représente la fonction d'activation ReLU et K le nombre de têtes d'attention. Ces têtes sont généralement utilisées pour gagner en stabilité et permettre d'obtenir des coefficients indépendants entre deux mêmes nœuds [Vas+17]. En absence de boucles, nous pouvons ajouter une connexion résiduelle permettant de véhiculer l'information passée portée par un nœud :

$$h_v^{(l)} = \tilde{h}_v^{(l)} + h_v^{(l-1)}. \tag{6.4}$$

Les connexions résiduelles (*residual connection* (RC) en anglais) [He+16 ; Dwi+20] sont généralement conçues pour faciliter l'apprentissage d'un réseau de neurones profond. Ce point sera discuté en profondeur dans la sous-section 6.2.2. Notons que tous les nœuds quel que soit le sous-graphe d'appartenance d'origine, sont mis à jour en parallèle.

La couche de réduction de *pooling* se base sur les méthodes top-r [Gra+21] cherchant à attribuer un score d'importance à chaque nœud et ne sélectionner que les r-nœuds de score le plus élevé. L'objectif ici est de ne garder que les nœuds du sous-graphe central de gènes $\mathcal{G}_G^*$ qui concentrent le plus d'information, dont une partie provient de leur voisinage. Pour cela, nous recherchons des nœuds dont les voisins sont également informatifs pour lesquels la représentation vectorielle a une norme élevée. Les sous-graphes d'annotations $\mathcal{G}_{\mathcal{K}_i}$ restent, quant à eux, inchangés. Par l'intermédiaire de ces sous-graphes d'annotations, les gènes vont pouvoir continuer à interagir au travers des nœuds dans ces sous-graphes même s'ils appartiennent à des composantes connexes différentes, palliant ainsi la perte de connectivité au sein du sous-graphe central $\mathcal{G}_G^*$. Un score s_v pour un gène v du sous-graphe central à la couche l est calculé de la manière suivante :

$$s_v^{(l)} = \alpha \times h_v^{(l)} + \frac{(1-\alpha)}{(B+1)} \times \Big(\max_{u\in\mathcal{G}_G^{*(l)}} \big(h_u^{(l)} | u \in \mathcal{N}_{intra}^{(l)}(v)\big) + \sum_j^B \max_{u\in\mathcal{G}_{\mathcal{K}_j}^{(l)}} \big(h_u^{(l)} | u \in \mathcal{N}_{inter}^{(j,l)}(v)\big)\Big) \tag{6.5}$$

où α est un hyperparamètre à déterminer. Nous désignerons par la suite cette approche par "local_max_intra&inter". La fonction "top" retournera les indices des $\lceil|\mathcal{V}_G^{(l)}| \times r^{(l)}\rceil$ nœuds ayant eu les meilleurs scores où $r^{(l)}$ désigne le ratio de sélection à la couche l. Dans le cas d'une représentation vectorielle à D-dimension, il faudra d'abord passer par une projection des représentations des nœuds avant de calculer ce score. Vu que la représentation vectorielle est à valeur dans $\mathbb{R}$ dans le cas des profils GE, il n'a pas été nécessaire de faire cette projection.

Un bloc est ainsi constitué d'une couche de convolution et d'une couche de pooling. À l'issue de chaque bloc, nous obtenons un sous-graphe central de taille réduite tel que $\mathcal{G}_G^{*(l)} = \text{top-}r(s^{(l)})$, $|\mathcal{V}_G^{(l)}| < |\mathcal{V}_G^{(l-1)}|$. Notons que les nœuds du sous-graphe central $\mathcal{G}_G^{*(0)}$ reçoivent initialement des gènes associés au signal issu du profil d'expression génétique (GE) d'un patient. Les nœuds des sous-graphes d'annotations sont pour leur part initialisés par une première propagation telle que

$$h_v^{(0)} = \begin{cases} GCONV(\hat{h}_v^{(0)}) \text{ où } \hat{h}_v^{(0)} = 0 & \text{si } v \notin \mathcal{V}_G^{(0)}, \\ x_v & \text{sinon.} \end{cases} \tag{6.6}$$

La couche de *readout* transforme enfin uniquement le sous-graphe central de gènes sous forme vectorielle afin de permettre de réaliser la tâche finale de prédiction. Cette couche concatène l'information issue du dernier bloc L de convolution-pooling provenant des gènes restants dans le sous-graphe central $\mathcal{G}_G^{*(L)}$, tout en conservant la trace des indices des gènes d'origine conservés jusqu'à ce dernier bloc. Pour ce faire, similairement à la couche de sélection dans GraphGONet, un vecteur M de dimension égale au nombre initial de gènes $|\mathcal{E}_G|$ est créé. Les entrées de ce

vecteur, correspondant aux derniers gènes sélectionnés, reçoivent le signal de ces derniers issu du bloc L tel que :

$$m_v = \begin{cases} h_v^{(L)} & \text{si } v \in \text{top-}r(s^{(L)}), \\ 0 & \text{sinon.} \end{cases} \tag{6.7}$$

Une couche linéaire dense suivie de la fonction softmax est finalement appliquée pour réaliser la tâche de classification finale.

6.2 Résultats

6.2.1 Formatage des graphes de connaissances

Nos expériences ont été menées jusqu'à présent seulement sur les données TCGA. On s'est intéressés à trois bases de connaissances : IPP, GO-BP, et Reactome, IPP formant le sous-graphe central et les deux derniers les sous-graphes d'annotations. Pour chaque ontologie ou réseau de connaissances, il a fallu procéder à un filtrage préliminaire pour ne retenir que les annotations avec les gènes contenus dans les données TCGA.

Dans le cas de IPP, le réseau contient initialement des interactions entre protéines. Il existe deux types d'interactions : physique directe et fonctionnelle. L'interaction physique connecte deux protéines issues par exemple d'un même complexe et peut les modifier. Ces changements peuvent être à l'origine de nouvelles fonctionnalités données aux protéines, par exemple pour assurer la phagocytose, processus d'ingestion de micro-organismes étrangers. Chaque interaction est munie d'un score de confiance variant de 0,15 à 0,9. Plus le score est faible, plus il y a d'interactions, mais moins elles sont fiables. Nous avons choisi de retenir les protéines impliquées dans des interactions où le score de confiance est élevé, supérieur ou égal à 0,7. Il a fallu ensuite procéder à une transformation de ce réseau de protéines en un réseau de gènes, la transformation étant admise, et enfin supprimer les gènes qui ne sont pas dans TCGA. Nous avons ainsi obtenu un graphe non orienté et non connexe $\mathcal{G}_G^*$ tel que $\mathcal{V}_G = 10947$ et $\mathcal{E}_{G,\text{intra}} = 115745$. Seuls 19,34 % des gènes initiaux sont retenus. Ce graphe est très parcimonieux avec une connectivité d'environ 0,19 %. Dans ce graphe, on comptabilise 86 composantes connexes, dont une principale contenant plus de 97,99 % de nœuds, de degré moyen 21,55. Les 85 restantes sont relativement de petite taille allant de sept à deux nœuds et de degré moyen 1,48. Nous avons choisi de garder les petites composantes puisqu'elles peuvent contenir de l'information intéressante pour la prédiction et aider à déduire de nouvelles connaissances. Certains gènes des petites composantes sont notamment liés aux gènes de la composante principale par l'intermédiaire des sous-graphes d'annotations. Les gènes de IPP non annotés ont été conservés puisqu'ils peuvent communiquer avec les autres gènes dans les composantes connexes.

Le graphe entier Reactome est initialement un DAG de 2546 nœuds et de 2565 arêtes. Contrairement à GO-BP, il n'est pas de faible connexité du fait qu'il dispose de 28 racines formant 19 composantes de faible connexité. Nous avons choisi de créer un nœud racine virtuel reliant les 28 racines entre elles pour permettre de faciliter la circulation de l'information provenant des gènes. Concernant GO-BP, nous nous sommes basés sur le graphe nettoyé et non tronqué utilisé dans GraphGONet (décrit à la sous-section 5.2.1 p. 70). Toutes les branches de ces graphes qui ne sont liées à aucun gène dans TCGA sont élaguées pour ne retenir que l'information essentielle et ainsi permettre aux données de tenir en mémoire. Si un gène est annoté avec un nœud des graphes d'annotations et les ancêtres de ce nœud, nous ne considérons pas les annotations avec les ancêtres. Nous obtenons ainsi pour GO-BP le graphe $\mathcal{G}_{GO}(\mathcal{V}_{GO} = 16\,062, \mathcal{E}_{GO,\text{intra}} = 37\,406)$ et pour Reactome le graphe $\mathcal{G}_P(\mathcal{V}_P = 2502, \mathcal{E}_{P,\text{intra}} = 2521)$. La majorité des niveaux de GO et Reactome sont connectés à au moins un gène. 92,63 % et 61,78 % des gènes de IPP sont annotés

respectivement par GO et Reactome. L'union de deux bases de connaissances annotent 93,17 % des gènes de $\mathcal{G}_G$. Notons que ces connaissances sont associées respectivement à 8286 et 3838 gènes supplémentaires du profil GE dans TCGA.

Nous pouvons alors construire deux versions de ces graphes : une première où on ne retient que les annotations avec les gènes contenus dans IPP et une seconde où les gènes non contenus initialement dans IPP sont ajoutés dans le graphe IPP $\mathcal{G}_G$. La Table 6.1 résume la connectivité des graphes de connaissances filtrés. Nous pourrions obtenir un recouvrement de 34,08 % des gènes initiaux dans TCGA en combinant IPP, GO-v2 et Reactome-v2. Notons que les relations entre GO-BP et Reactome n'ont pas été retenues. Certaines racines de Reactome représentent des termes GO de niveau moyen 4. D'autres relations pourraient être extraites à l'aide d'outils bio-informatiques. Néanmoins, le principal intérêt d'utiliser les connaissances auxiliaires est de faire communiquer les gènes, même éloignés dans le sous-graphe central. Ces connaissances auxiliaires pourront aider à l'interprétation.

Source	Taille	$\mathcal{V}$	$\mathcal{E}_{\text{intra}}$	avg(d_{intra})	$\mathcal{E}_{\text{inter}}$	#gènes_annotés (avg±std/node)
GO-BP						
v1	H=20	14 685	34 047	$4,64 \pm 17,51$	73 580	10 141 $(5,01 \pm 15,24)$
						smallcc : 130, bigcc : 10 011
v2	H=20	16 062	37 406	$4,66 \pm 18,70$	116 926	18 427 $(7,28 \pm 25,00)$
						+ isolés : 8286
Reactome (P)						
v1	H=12	2399	2416	$2,01 \pm 5,66$	29 305	6763 $(12,22 \pm 19,86)$
						smallcc : 42, bigcc : 6721
v2	H=12	2502	2521	$2,02 \pm 6,81$	43 583	10 601 $(17,42 \pm 29,87)$
						+ isolés : 3838
IPP (G)						
v1	CC=86	10 947	115 745	$21,15 \pm 978,59$	102 885 (GO-v1+P-v1)	10 199 $(10,09 \pm 13,29)$
						smallcc : 134, bigcc : 10 065
v2	CC=86	19294	115 745	$12,00 \pm 555,23$	160 509 (GO-v2+P-v2)	18546 $(8,65 \pm 12,04)$
						+ isolés : 8347

Table 6.1 – ***Description des graphes de connaissances (GO-BP, Reactome, IPP) formatés.*** *La colonne "Source" indique les versions éditées des connaissances utilisées. La colonne "Taille" indique en fonction du type de graphe soit le nombre de composantes connexes (CC), soit la hauteur du graphe (H) correspondant à la distance maximale entre un nœud et la racine. Les colonnes $\mathcal{V}$, $\mathcal{E}_{intra}$ et avg(d_{intra}) donnent des informations sur le nombre de nœuds, d'arêtes intra-communautaires et le degré moyen associé. Les colonnes $\mathcal{E}_{inter}$ et #gènes_annotés (avg±std par nœud) décrivent le nombre d'arêtes inter-communautaires et le nombre de gènes annotés par les graphes auxiliaires (la moyenne et l'écart-type par nœud). Ces gènes peuvent provenir soit de la composante connexe principale ("big-cc") ou des petites composantes connexes ("small-cc") de IPP. Le qualificatif "isolés" représentent des gènes de TCGA initialement non-contenus dans IPP mais qui y ont été ajoutés, car annotés par Reactome ou GO-BP.*

6.2.2 Analyse de sensibilité

Différentes expériences ont été menées pour évaluer l'efficacité de GraphGONet sur un problème de classification multiclasses. Nous avons conservé l'optimiseur Adam avec un taux d'apprentissage adaptatif de 0,001. La taille de batch a été réduite à 16, principalement pour mieux gérer la mémoire. L'apprentissage est également contrôlé par l'arrêt prématuré avec une patience de 5 et un delta de 0,001. La conception de BioHAN a demandé un travail plus approfondi sur l'ajustement des hyperparamètres. Cet ajustement a été effectué sur l'ensemble de validation

tandis que l'évaluation finale du modèle s'est faite sur l'ensemble de test, une fois les hyperparamètres fixés. La métrique de performances ici utilisée est l'accuracy. L'ensemble des expériences décrites par la suite ont été conduites sur des serveurs équipés de GPU RTX 2080Ti et A6000 dans un environnement PyTorch v1.11.0 et PyTorch Geometric v2.0.4.

Dans la suite, nous allons reporter les principaux résultats issus de l'ajustement des hyperparamètres. Puis, nous explorons la sensibilité du modèle en fonction de la connaissance et du nombre d'exemples d'apprentissage utilisés par rapport à l'état de l'art.

i) Ajustement des hyperparamètres Afin de définir la valeur des hyperparamètres du modèle, nous avons mené les expériences sur le graphe $\mathcal{G}_G$ annoté par $\mathcal{G}_{GO-v1}$ et sur l'ensemble d'apprentissage entier. Pour certains hyperparamètres, nous nous sommes appuyés des valeurs communément prises dans la littérature dans le but de réduire l'espace de recherche. De cette façon, la valeur de l'hyperparamètre ns de la fonction LeakyReLU a été fixée à 0.2. Concernant la couche convolution, nous nous sommes intéressés à la manière de prendre en compte l'information passée d'un nœud lors de sa mise à jour et à la variation du nombre de têtes d'attention. En ce qui concerne le premier point, l'information passée du nœud est habituellement résumée dans les GNNs par la fonction COMBINE (Eq. (2.1b)). Cela passe généralement par l'intégration du nœud dans son voisinage comme dans GCN [KW17] ou GAT [Vel+18]. On parle alors de boucle (*self-loop* (SL) en anglais). Une alternative est de recourir aux connexions résiduelles comme formulée par l'Eq. (6.4). L'étude conduite par Dwivedi et al. [Dwi+20] avait montré que les connexions résiduelles sont à préférer par rapport aux boucles, elles permettent de réduire le problème de la dissipation du gradient pendant la rétropropagation et d'augmenter la profondeur du réseau. Un gain de performances est souvent constaté. Nous avons de ce fait évalué la combinaison des deux selon le schéma suivant : {NO_RC & NO_SL, RC & NO_SL, NO_RC & SL, RC & SL}. Les connexions résiduelles sont ajoutées entre chaque bloc, tandis que les boucles sont intégrées à chaque convolution. Cinq modèles de BioHAN à deux blocs de convolution-pooling ont été appris sur chaque configuration. Les résultats sont reportés dans la Table 6.2a. On constate en effet que les meilleures performances sont acquises en présence seules des connexions résiduelles (configuration RC & NO_SL). Le nœud central est traité différemment par rapport aux nœuds du voisinage et la taille du graphe ne s'en retrouve pas augmentée. Malgré que la combinaison RC & SL affiche une accuracy moyenne de 0,961, cela ne permet pas d'améliorer plus les performances. Nous avons donc configuré le modèle de sorte qu'il intègre des connexions résiduelles entre chaque bloc et aucune boucle. Nous avons ensuite cherché à faire varier le nombre de têtes d'attention de un à six, suivant le même protocole. Les résultats sont présentés dans la Table 6.2b. On observe que les performances du modèle sont aussi bonnes qu'avec une ou deux têtes. Notons que l'ajout d'une tête d'attention supplémentaire s'accompagne d'une augmentation des paramètres. Pour moins complexifier le modèle, le nombre de têtes d'attention K a donc été fixé à un.

a)

	NO_RC & NO_SL	RC & NO_SL	NO_RC & SL	RC & SL
ACC (avg±std)	$0,596 \pm 0,389$	$0,964 \pm 0,002$	$0,609 \pm 0,401$	$0,961 \pm 0,008$

b)

Nombre de têtes	1	2	3	4	6
ACC (avg±std)	$0,964 \pm 0,004$	$0,964 \pm 0,002$	$0,962 \pm 0,004$	$0,960 \pm 0,010$	$0,962 \pm 0,009$

Table 6.2 – ***Résultats des expérimentations menées pour définir la valeur des hyperparamètres de la couche de convolution de BioHAN.***

Au sujet de la couche de pooling, nous avons évalué la méthode de calcul du score $s^{(l)}$ d'un nœud donnée par l'Eq. (6.5) ainsi que le ratio de pooling r. Concernant l'évaluation du score $s^{(l)}$ appris par la méthode "local_max_intra&inter", nous avons obtenu de meilleures performances en fixant l'hyperparamètre α à 0.5. Nous avons comparé les résultats de cette méthode à ceux obtenus par d'autres approches de l'état de l'art [Hua+19 ; LLK19], résumées dans la Table 6.3a. La méthode "global_softmax" calcule l'importance d'un nœud vis-à-vis de l'ensemble des nœuds du graphe étudié, tandis que la méthode "local_softmax" évalue l'importance d'un nœud par rapport à son voisinage *intra*. Dans les deux cas, le score obtenu est normalisé par la fonction softmax. Les auteurs préconisent de prendre en compte la taille du voisinage dans le calcul du score final de la méthode "local_softmax" pour éviter les biais liés à celle-ci. La méthode SAGPool proposée par Lee et al. [LLK19] consiste à calculer un score d'auto-attention par une couche de convolution et d'utiliser ce score comme métrique de classement dans la fonction top-r. Cette méthode a le désavantage parfois de ne pas considérer des régions entières du graphe lors de la sélection [Gra+21] et, dans certains cas extrêmes, de réduire le graphe à un ensemble de nœuds isolés [GLJ21]. La méthode "local_max_intra" est une variante de la méthode de calcul proposée qui ne considère que le voisinage *intra*-communautaire d'un nœud. L'évaluation s'est faite dans les mêmes conditions que précédemment dont les résultats sont donnés dans la Table 6.3b. Nous ne remarquons pas d'écart net de performances, excepté une perte d'accuracy de 0,7 à 0,8 % entre "global_softmax" et les autres méthodes. Malgré une légère perte de 0,1 % avec notre méthode, nous avons observé que les nœuds sélectionnés gardent plus de connectivité et sont davantage liés aux nœuds des sous-graphes d'annotations qu'avec "local_softmax". SAGPool représente un bon candidat, mais rend l'apprentissage plus long du fait de l'apprentissage de paramètres supplémentaires. De plus, cette approche présente plus d'intérêts dans un espace de représentation de plus grande dimension. Enfin, la méthode "local_max_intra" a tendance à sélectionner les nœuds de forte connectivité, appartenant essentiellement à la composante connexe principale. Notre approche représente alors un bon compromis.

a)

local_softmax [Hua+19]	global_softmax [Hua+19]	local_max_intra
$\frac{\exp(h_v^{(l)})}{\sum_{u \in \mathcal{N}^{(l)}_{G-G^{(v)}}} \exp(h_u^{(l)})} \times \lvert\mathcal{N}^{(l)}_{G-G}(v)\rvert$	$\frac{\exp(h_v^{(l)})}{\sum_{u \in \mathcal{V}_G} \exp(h_u^{(l)})}$	$\alpha h_v^{(l)} + (1-\alpha) \times \max\left(h_u^{(l)} \lvert u \in \mathcal{N}^{(l)}_{intra}(v)\right)$

b)

méthode	local_softmax	global_softmax	SAGPool	local_max_intra	local_max_intra&inter
ACC	$0,965 \pm 0,004$	$0,957 \pm 0,004$	$0,965 \pm 0,004$	$0,965 \pm 0,002$	$0,964 \pm 0,002$

Table 6.3 – ***Résultats des expérimentations menées pour définir la valeur des hyperparamètres de la couche de réduction de BioHAN.***

Le ratio r de la couche de pooling dépend fortement du nombre blocs de convolution-pooling appliqués. L'objectif principal de l'usage de plusieurs blocs est de décroître au fur et à mesure la taille du sous-graphe central initialement à 10,9K nœuds pour obtenir une taille réduite raisonnable, suffisamment interprétable. Les modèles standards GNNs sont généralement peu profonds, pas plus de cinq couches de convolution. Notre approche a été testée sur une profondeur maximale de trois. Nous avons remarqué qu'en maintenant la même valeur de ratio, les performances se dégradaient. Après plusieurs tests, dans une configuration à deux blocs, nous avons fixé $r^{(1)} = 0.5$ et $r^{(2)} = 0.1$ et dans une configuration à trois blocs, $r^{(1)} = 0.5$, $r^{(2)} = 0.5$ et $r^{(3)} = 0.5$. On obtient alors des sous-graphes centraux respectivement de taille finale 548 et 274. On constate une légère baisse de performances d'environ 0,7% avec une configuration à trois blocs. Nous avions également testé avec une couche de convolution finale après le dernier bloc et avant la couche de

readout. Les performances restaient inchangées. Une architecture à deux blocs suivie directement d'une couche readout a donc été favorisée dans les prochaines expériences. La Table 6.4 résume les hyperparamètres évalués et les valeurs choisies sur la base de cette étude.

Hyperparamètres	Valeur choisie	Valeurs testées
L	2	[1,2,3]
K	1	[1,2,3,4,6]
ns	0,2	-
τ	1,0	-
RC/SL	RC&NO_SL	(NO)_RC&(NO)_SL
α	0,5	[0,4-0,6]
r	0,5($l = 1$) - 0,1($l = 2$)	[0,001-0,5]

Table 6.4 – ***Liste des hyperparamètres optimisés durant la phase d'apprentissage.***

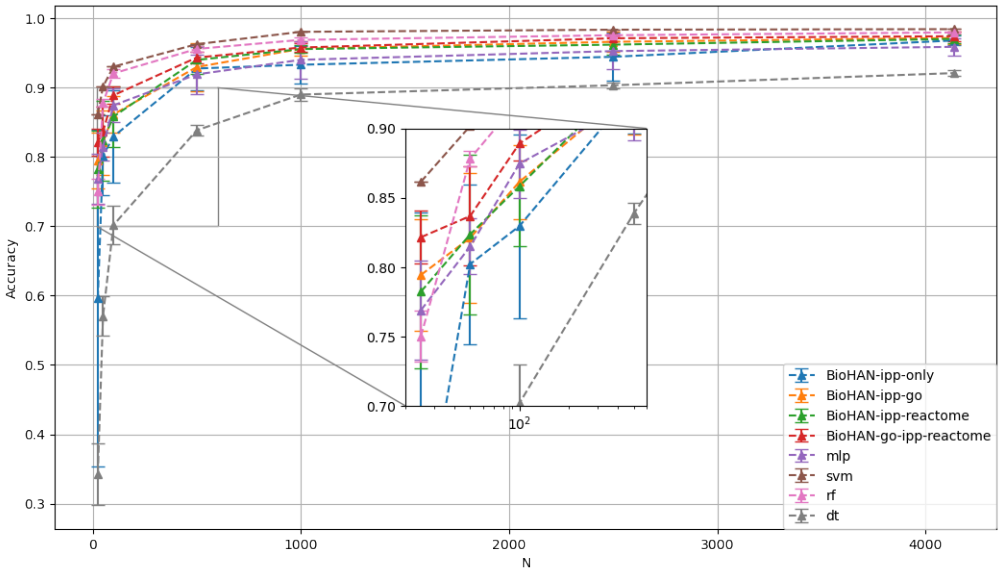
ii) Comparaison avec l'état de l'art selon deux variations Dans cette seconde série d'expériences, nous comparons différentes variantes de BioHAN en fonction des graphes de connaissances utilisés et du nombre d'exemples d'apprentissage. Cette comparaison est menée par rapport à des méthodes ML de l'état de l'art. Concernant les graphes de connaissances exploités, nous avons considéré différentes alternatives :

— une première où seul le sous-graphe central de gènes $\mathcal{G}_G^*$ est considéré, correspondant ainsi à un graphe homogène (BioHAN-ppi-only) ;
— deux autres où le graphe hétérogène $\mathcal{G}$ est formé de $(\mathcal{G}_G^*, \mathcal{G}_{\mathcal{K}_1})$ avec $\mathcal{K}_1 = GO$ ou P (BioHAN-ppi-go, BioHAN-ppi-reactome) ;
— une dernière où $\mathcal{G} = (\mathcal{G}_G^*, \mathcal{G}_P, \mathcal{G}_{GO})$ (BioHAN-go-ppi-reactome).

Les méthodes ML de l'état de l'art sont les mêmes que précédemment, à savoir un DT (critère de Gini), RF (critère de Gini, nombre d'arbres=100), SVM (noyau linéaire, C=1,0) et MLP (trois couches avec respectivement 1000, 500 et 200 neurones). Différentes bases de gènes peuvent être considérées en fonction de la proportion de gènes TCGA ajoutés au sous-graphe central $\mathcal{G}_{G\text{-v1}}^*$, initialement non présents dans le graphe IPP, mais annotés par les graphes auxiliaires de connaissances. Nous avons pu évaluer les méthodes sur deux bases différentes de gènes respectivement de taille 10 947 et 19 233. Cette différence représente la part de gènes TCGA complémentaires annotés par le sous-graphe $\mathcal{G}_{GO\text{-v2}}$. Ces gènes additionnels sont représentés par des nœuds isolés dans le sous-graphe central $\mathcal{G}_{G\text{-v1}}^*$. Nous avons donc procédé à des évaluations distinctes des méthodes ML de l'état de l'art et BioHAN sur ces deux bases de gènes. Par manque de temps, nous avons seulement réévalué BioHAN-ppi-only et BioHAN-ppi-go sur la seconde base de gènes. Néanmoins, ces premiers résultats permettront de juger de l'apport de la connaissance externe face à l'injection de nœuds isolés dans le graphe d'entrée. Nous faisons également varier la taille de l'ensemble d'apprentissage entre 25 et 4136. Dix modèles ont été appris pour chaque taille d'échantillon. Les courbes d'accuracy sont présentées dans les Fig. 6.2(a-b) en fonction du nombre d'exemples d'apprentissage et de la base de gènes considérée.

Tout d'abord, nous notons que globalement les modèles arrivent à bien apprendre en présence d'un nombre suffisant d'échantillons d'entraînement (à partir de 1000). Les performances commencent à se dégrader progressivement en-dessous de 500 exemples, quel que soit le modèle. Les méthodes boîtes noires SVM et RF sont légèrement plus performantes que les différentes variantes de BioHAN alors que le MLP, une autre méthode boîte noire, a une accuracy généralement en dessous. Notre approche reste ainsi aussi compétitive que ces méthodes. Les résultats obtenus avec la variante BioHAN-ppi-only incluant seulement le graphe IPP sont en général moins bons qu'avec les autres variantes proposées et sont similaires au MLP. Les courbes de ces deux modèles ont tendance à s'entrecroiser. Ce constat est d'autant plus marqué sur la Fig. 6.2a. La connaissance auxiliaire aurait tendance à mieux faire circuler l'information au sein du réseau de neurones et permettre alors au modèle de résoudre la tâche de classification plus efficacement. Nous constatons de plus que sur la Fig. 6.2a, la variante BioHAN-go-ppi-reactome s'avère être la plus compétitive parmi les différentes variantes évaluées et dispose d'une différence de performances inférieure à 0,4 % en moyenne par rapport au SVM. Il apparaît aussi que plus nous disposons de connaissances auxiliaires, plus nous parvenons à faire communiquer les gènes même distants, meilleures sont les performances. Sur la Fig. 6.2b, le gain d'accuracy de BioHAN-ppi-go par rapport à BioHAN-ppi-only ne dépasse pas les 1 %. Sur une taille d'échantillons de 25, BioHAN-ppi-only a une accuracy légèrement supérieure en moyenne. L'ajout de connaissances supplémentaires pourrait aider à régulariser le modèle BioHAN-ppi-go. Toutefois, nous observons que pour des valeurs d'échantillon de 25 et 50 exemples d'apprentissage, certains modèles de BioHAN-ppi-go et BioHAN-ppi-only parviennent à faire mieux qu'un SVM. De toute évidence, l'ensemble des variantes BioHAN surpassent nettement l'arbre de décision, la seule méthode interprétable par essence.

a) $|\mathcal{V}_G| = 10947$



b) $|\mathcal{V}_G| = 19233$

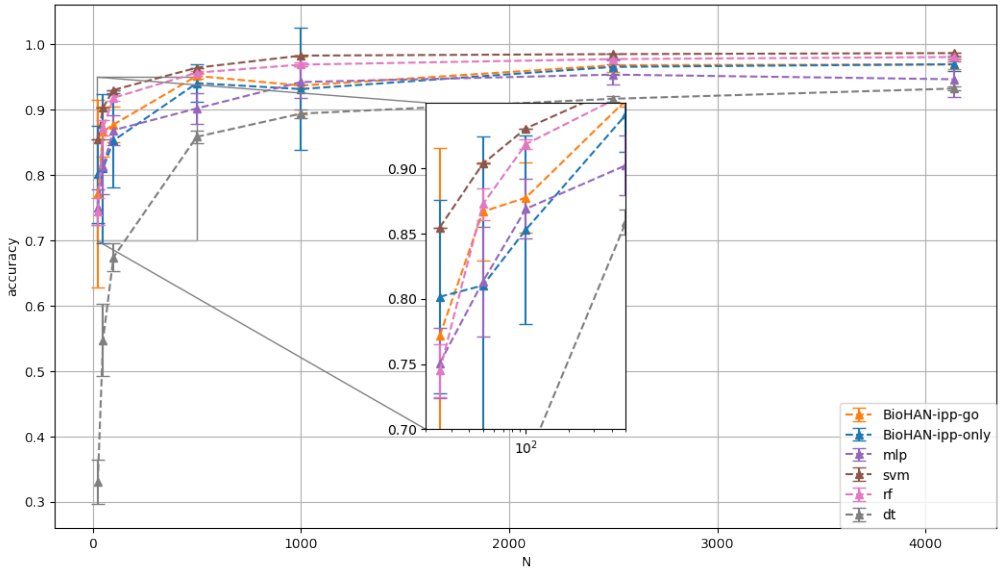


FIGURE 6.2 – ***Accuracy des modèles selon (a-b) la base de gènes utilisée et le nombre d'exemples d'apprentissage N sur le jeu de données TCGA.***

6.2.3 Explication biologique de la prédiction d'un patient

Nous donnons dans cette section un aperçu de l'explication individuelle produite par cette nouvelle méthode. L'objectif reste le même que les précédentes méthodes, c'est-à-dire la production d'explications fiables et intelligibles basées sur la connaissance à partir desquelles il est possible d'identifier les concepts biologiques qui ont été les plus mobilisés dans le calcul de la prédiction. BioHAN a l'avantage de dresser automatiquement l'ensemble réduit des gènes utilisés pour la tâche de prédiction. Tout comme dans GraphGONet, nous pouvons discriminer les plus importants à l'aide de la méthode *Gradients*×*Inputs* (GI). À partir de ce sous-ensemble, il est possible de retracer a posteriori le parcours emprunté par le signal au travers des sous-graphes de connaissance en inspectant les scores d'attention et de pooling et ainsi générer des graphes d'explication. Le processus est résumé par la Fig. 6.3.

Nous proposons par la suite d'analyser la prédiction d'un des modèles de BioHAN-go-ppi-

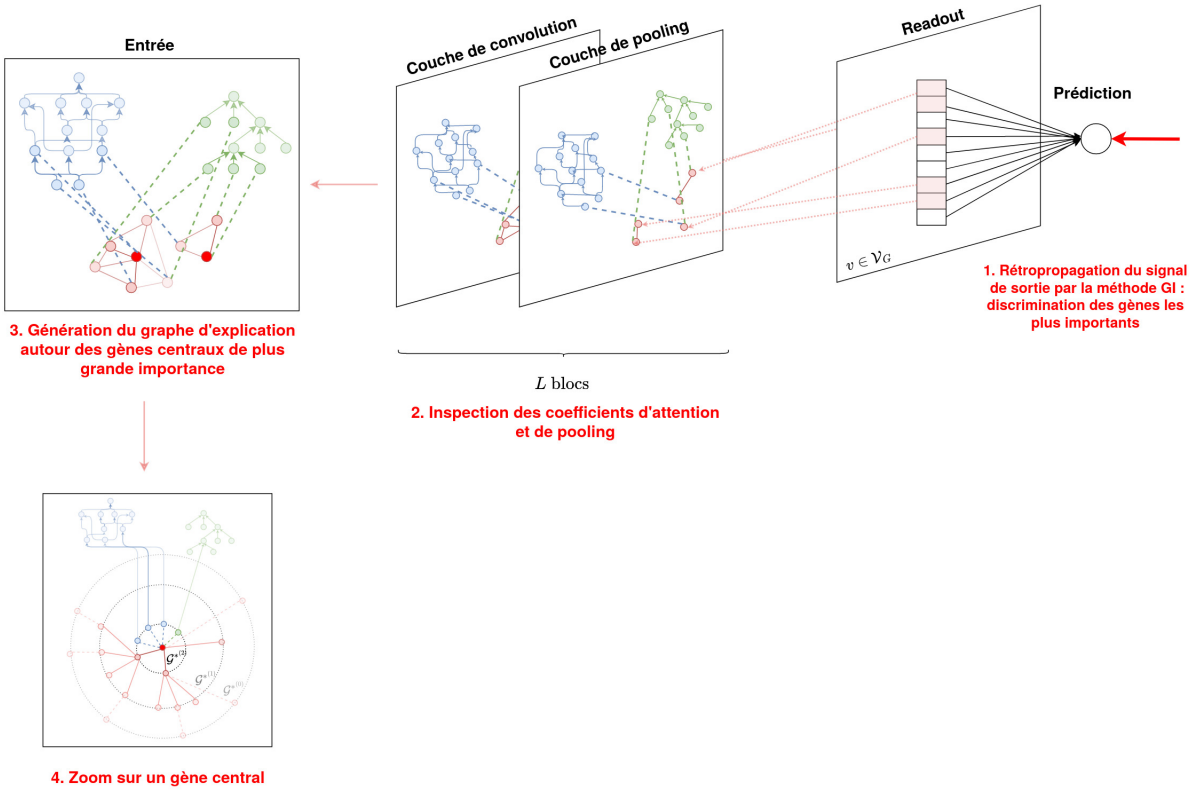
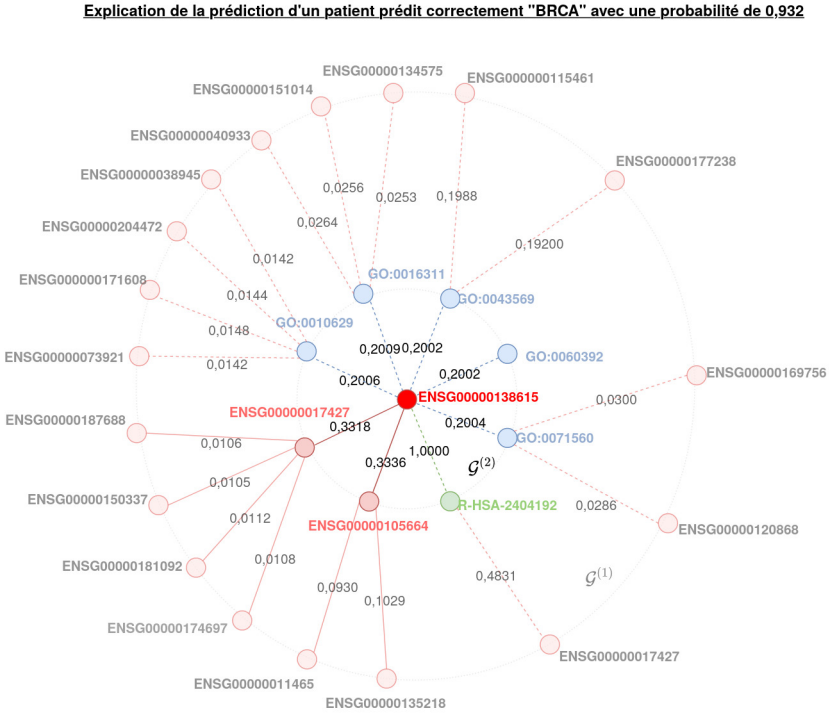


FIGURE 6.3 – ***Processus de la génération des sous-graphes d'explication.***

reactome appris précédemment sur l'ensemble complet d'apprentissage. Les ensembles finaux de gènes générés par ce modèle contiennent 548 nœuds. Les gènes retenus dans ces ensembles seront différents pour chaque patient. Pour un patient donné, nous calculons sur l'ensemble de gènes finaux leur score de pertinence afin d'établir un classement des gènes les plus importants. Nous rappelons que dans le cas d'une sortie softmax, plus le score est élevé envers une classe, plus le gène aura un impact positif sur celle-ci. Nos analyses ont montré que la distribution de ces scores suit une loi gaussienne centrée en zéro. En supposant que ces scores suivent une distribution $\mathcal{N}(0, \sigma)$ où σ est la variance empirique, nous pouvons recourir à un test-t bilatéral avec une valeur-p fixée à 0,05 pour déterminer les scores éloignés de la borne supérieure et ainsi identifier les gènes les plus importants associés. De là, il est possible de reconstruire les sous-graphes d'interactions centrés sur ces gènes en identifiant les voisins avec qui l'attention est la plus forte. Nous présentons dans la Fig. 6.4 un exemple d'explication de la prédiction d'un patient issu de l'ensemble de test de TCGA correctement prédit "BRCA" avec une probabilité de 0,932 et un score de pertinence total de -0,1216. Un tableau y indique les quinze nœuds les plus importants qui sont ressortis par le t-test. Nous avons examiné le graphe d'explication construit autour du nœud central associé au gène le plus important "ENSG00000138615". Les nœuds du premier cercle représentent ses voisins *intra* et *inter*-communautaire d'ordre 1 dans le graphe réduit $\mathcal{G}^{(2)}$. La pondération sur les arcs indique le score d'attention $\alpha_{uv}^{(2,1)}$ obtenu lors de la couche de convolution du second bloc précédent le pooling. Les nœuds du second cercle représentent, quant à eux, les voisins du nœud central d'ordre 2 récupérés dans le graphe réduit $\mathcal{G}^{(1)}$, c'est-à-dire issu du premier bloc de convolution-pooling. La pondération est le score d'attention $\alpha_{uv}^{(1,1)}$ calculé lors de la première couche de convolution. Le voisinage de ces nœuds pouvant être particulièrement grand, par soucis de clarté, nous avons préféré ne choisir que certains voisins. Ces voisins ont été triés selon leur score d'attention : les nœuds dont le voisinage a une taille supérieure à 10, seuls les voisins du troisième quartile ont été retenus. Lorsque la taille du voisinage est supérieure à

50, on a retenu ceux du quartile 0.95. On observe que l'attention portée envers le voisinage est variable. Nous pouvons identifier de la sorte les nœuds qui ont influencé l'information portée par le nœud central. Par exemple, le gène central "ENSG00000138615" porte une plus grande attention dans son voisinage d'ordre 1 au gène "ENSG00000105664" qu'au gène "ENSG00000017427". Ce gène central est également influencé indirectement au travers du terme "GO :0043569" par les gènes "ENSG00000115461" et "ENSG00000177238" composant son voisinage de second ordre. Les valeurs d'attention y sont relativement élevées par rapport aux autres valeurs d'attention dans le voisinage du même ordre. Nous avons pu aussi noter que les graphes réduits construits par le réseau sont généralement formés d'une composante connexe principale sur la base du graphe d'origine et de quelques composantes de taille relativement faible (1 à 5 nœuds). Majoritairement, ces dernières restent connectées à la composante connexe principale par l'intermédiaire d'un terme GO ou Reactome.

Ce type de sous-graphe d'explication permet de ce fait de rendre compte des gènes importants pour la prédiction et ceux qui les ont influencés. Les connaissances auxiliaires viennent aussi compléter ces explications pour mieux caractériser ces gènes. Des tests statistiques d'enrichissement biologique ne sont donc plus nécessaires. Ces connaissances auxiliaires apparaissent également comme des voies de communication avec des gènes distants ou isolés dans des composantes connexes de petite taille. On dispose ainsi d'une vision plus large des différentes interactions entre ces concepts biologiques (gène, fonction biologique, voie métabolique) qui sont compréhensibles par des utilisateurs familiers avec ces connaissances. Les biologistes peuvent facilement se les approprier et en valider l'explication si elle a un sens avec le phénotype étudié. Par exemple, le gène central "ENSG00000138615" définit l'hormone peptidique IGF-1, littéralement facteur de croissance 1 ressemblant à l'insuline de l'anglais *insulin-like growth factor-1*. L'activité cellulaire liée est précisée par les termes GO et Reactome voisins tels que la voie métabolique "R-HSA-2404192" caractérisant la voie de signalisation liée à ce facteur de croissance et le terme "GO :0043569" associé à la régulation négative de ce facteur. Des études ont montré que cette hormone de croissance, à des concentrations élevées, pourrait favoriser le développement de cancer, dont le cancer du sein [YR00 ; Mur+20].



Classement des gènes les plus importants (test-t bilatéral; valeur-p=0,05)	
Identifiant gène	**Score de pertinence**
ENSG00000138615	0,1208
ENSG00000211455	0,0971
ENSG00000181092	0,0916
ENSG00000164754	0,0906
ENSG00000139187	0,0889
ENSG00000185697	0,0856
ENSG00000273049	0,0802
ENSG00000178789	0,0774
ENSG00000116996	0,0765
ENSG00000143387	0,0748
ENSG00000196664	0,0690
ENSG00000009790	0,0654
ENSG00000186141	0,0642
ENSG00000183160	0,0626
ENSG00000005302	0,0606
Score de pertinence total	-0.1216

Zoom sur le graphe d'explication généré autour du gène le plus important "ENSG00000138615"

Identifiant	Libélé
ENSG00000138615	*insulin like growth factor 1*
ENSG00000138615	*cartilage intermediate layer protein*
ENSG00000105664	*cartliage oligomeric matrix protein*
GO:0060392	*negative regulation of SMAD protein signal transduction*
GO:0043569	*negative regulation of insulin-like growth factor receptor signaling pathway*
GO:0016311	*dephosphorylation*
GO:0010629	*negative regulation of gene expression*
GO:0071560	*cellular response to transforming growth factor beta stimulus*
R-HSA-2404192	*signaling by type 1 insulin-like growth factor 1 receptor*

FIGURE 6.4 – ***Explication de la prédiction d'un patient prédit correctement "BRCA" avec une probabilité de 0,932.*** *Le tableau de droite indique le classement des gènes les plus pertinents suite au test-t de valeur-p 0,05. Un sous-graphe à gauche a été généré autour du nœud central associé au gène le plus important "ENSG00000138615". Les nœuds du premier cercle représentent ces voisins d'ordre 1 et les nœuds du second cercle ceux d'ordre 2. La pondération sur les arcs représente les scores d'attention entre chaque paire de nœuds.*

6.3 Discussion et conclusion

Les premiers résultats de cette approche sont prometteurs. L'analyse de sensibilité a montré que le modèle enrichi de plusieurs bases de connaissances fait mieux qu'un réseau de neurones boîte noire comme un MLP. En plus, d'être compétitif avec des méthodes ML boîtes noires de l'état de l'art, ce modèle a l'avantage d'être interprétable par construction et d'être aussi *self-explaining* que l'est GraphGONet. Une explication de la prédiction d'un patient a été proposée sous la forme d'un sous-graphe centré sur le gène le plus important. L'explication est facilement à la portée de différents utilisateurs familiers avec la connaissance employée.

BioHAN se distingue de l'état de l'art sur deux principaux points. D'abord, dans la littérature relativement limitée des GNNs sur les graphes hétérogènes, des chaînes dites "metapath" sont employées pour extraire de nouvelles relations entre des nœuds de distance deux qui originalement n'interagissent pas ensemble. Cependant, à notre connaissance, peu de graphes hétérogènes ont une configuration en forme d'étoile, c'est-à-dire avec un sous-graphe central autour duquel gravitent des sous-graphes auxiliaires servant à assurer sa connectivité tout le long du modèle. Ensuite, les GNNs sur les graphes hétérogènes sont généralement conçus pour résoudre des tâches de classification de nœud et non de graphe sans avoir recours aux blocs de convolution-pooling et restent limités à une couche à deux couches de convolution. L'avantage d'utiliser des blocs est de permettre de réduire progressivement la taille du graphe et d'en extraire l'information la plus pertinente pour le problème. Au risque de créer plus de composantes connexes qui ne communiqueraient pas entre elles, il a été choisi de ne réduire que le graphe central pour permettre de toujours véhiculer l'information provenant des gènes. Ces graphes auxiliaires servent surtout à maintenir la connexité du graphe central parcimonieux et la communication d'information. Afin de garder la trace des gènes sélectionnés qui varient d'un patient à l'autre, nous avons repris l'idée de la couche de sélection dans GraphGONet pour formuler la couche de readout. Malgré la présence supplémentaire de paramètres dans les couches cachées de BioHAN et la gestion de plus de connaissances, en basculant totalement sur un GNN, BioHAN présente au minimum 10% de moins de paramètres que GraphGONet. À défaut d'avoir plusieurs blocs rendant le modèle plus complexe et ainsi pouvant défavoriser son interprétation, la technique de l'attention permet en contrepartie de guider l'apprentissage et faciliter l'interprétation a posteriori. Les explications produites par BioHAN se retrouvent enrichies biologiquement par rapport à celles de Deep GONet et de GraphGONet en replaçant le gène au centre.

Ce travail reste une première ébauche avant soumission. Quelques points restent à améliorer. Tout d'abord, nous espérons rendre les scores d'attention plus parcimonieux en jouant sur l'hyperparamètre de température pour mieux distinguer les voisins les plus influents et ainsi faciliter l'interprétation. La méthode de score utilisée par le pooling pourrait être aussi ajustée lorsque la connaissance utilisée est réduite au seul graphe IPP. Ensuite, nous prévoyons de mener des expériences additionnelles sur d'autres jeux de données pour valider ces premiers résultats. Nous souhaitons notamment appliquer ce modèle à des problèmes plus difficiles tels que la survie pour évaluer l'apport de la connaissance dans ce type de contexte. Il est prévu également que ce modèle soit testé sur la cohorte d'Oncodesign dans un contexte réel de peu de données. À propos de l'interprétation biologique des prédictions, l'interprétation locale proposée doit être affinée en projetant les graphes d'explications dans le graphe d'origine, et en inspectant davantage les coefficients d'attention à chaque bloc de convolution-pooling. Une interprétation globale du modèle reste aussi à construire. Nous examinerons la consistance des signatures génétiques et évaluerons si le fait d'avoir guidé l'apprentissage avec plusieurs connaissances permet de garantir cette stabilité même en centralisant l'information sur les gènes. Concernant la connaissance utilisée, nous pourrions jouer sur la valeur du score de confiance fixé dans le réseau IPP afin d'intégrer plus de gènes, pas forcément annotés par les graphes d'annotations. De plus, dans les données TCGA, il existe des parties non codantes du génome qui pour certaines sont annotées par des ontologies telles que GO. Elles pourraient être ainsi ajoutées au sous-graphe central en tant que

nœuds isolés, de la même manière que les gènes TCGA initialement non inclus dans IPP mais annotés à la connaissance. Enfin, il est possible de recourir à d'autres graphes de connaissances, peu importe leur topologie, à condition que cela passe en mémoire. La taille relative actuelle d'un graphe patient occupe de 3 à 8 Mo.

Chap.

7 Conclusion et perspectives

7.1 Conclusion

Ces trois années de thèse ont permis de développer de nouveaux modèles d'apprentissage profond originaux, interprétables par construction, pour la médecine de précision. Nous nous sommes en particulier intéressées aux données d'expression de gènes qui informent de l'état cellulaire des patients. La régulation de l'expression génétique intervient dans des mécanismes centraux comme la différentiation cellulaire, l'apoptose... L'étude de ces données par les algorithmes d'apprentissage profond permet d'acquérir les connaissances utiles pour mieux comprendre notamment leur rôle dans les maladies complexes telles que le cancer. Trois réseaux de neurones guidés par les connaissances biologiques ont été conçus, à savoir :

- Deep GONet, un perceptron multicouche totalement connecté dont les couches cachées représentent une partie de l'ontologie de gènes GO (décrit au chapitre 4) ;
- GraphGONet, un réseau de neurones combinant un réseau à propagation avant et un GNN intégrant les différents types d'interactions et degrés de spécificité de GO (décrit au chapitre 5) ;
- BioHAN, un GNN appliqué sur des graphes patients pouvant inclure différentes connaissances dont IPP, GO et Reactome (décrit au chapitre 6).

Chaque méthode a été évaluée sur des jeux de données publics pour des tâches portant sur le diagnostic médical. Nous avons réussi à montrer qu'elles étaient aussi compétitives que les méthodes ML opaques de l'état de l'art (SVM, RF, MLP). Nos méthodes se distinguent de cet état de l'art, d'une part du fait que l'architecture du réseau de neurones est compréhensible, notamment par des experts qui ont l'habitude de travailler avec cette connaissance et, d'autre part, par le fait que certaines d'entre elles (GraphGONet et BioHAN) sont *self-explaining*, c'est-à-dire qu'elles sont capables de rendre automatiquement des explications.

Nos méthodes s'inscrivent néanmoins dans la continuité de certaines familles de méthodes présentées dans le chapitre 3 sur l'état de l'art de l'interprétation des réseaux de neurones et de l'intégration des connaissances. Elles sont assez similaires sur le principe des méthodes à base de concepts telles que la méthode *self-explaining ConceptBottleneck* [Koh+20a] où l'objectif est d'extraire à partir des données des concepts sémantiques compréhensibles sur lesquels le modèle s'appuierait pour faire ses prédictions et qui pourraient ainsi servir de support d'explication. Au lieu de disposer de seulement d'une couche où les neurones représentent des concepts, nous l'avons étendu à plusieurs couches dans Deep GONet et GraphGONet. Les méthodes FFNN contraintes par la connaissance se fondent également sur ce principe, mais ne sont pas *self-explaining*. Certaines intègrent plusieurs couches de connaissances comme BioVNN [LL21] et ParsVNN [Hua+21]. Deep GONet s'en différencie par le fait d'avoir intégré les gènes non annotés et contraint les poids de couches totalement connectées par un terme de régularisation adaptatif. Cette méthode n'est pas considérée de *self-explaining* puisque nous avons eu recours à une méthode d'interprétation a posteriori pour identifier les neurones les plus pertinents vis-à-vis de la prédiction et leurs concepts biologiques associés. Cependant, le fait d'avoir associé a priori à un neurone un concept biologique, l'interprétation en est facilitée. Un test d'enrichissement

statistique GO n'est en effet plus nécessaire.

Quant à la méthode GraphGONet, elle a la particularité de prendre en compte une représentation sémantique plus riche de GO, c'est-à-dire des connexions entre couches non adjacentes et différents degrés de spécificité des niveaux GO. Cette intégration se fait au moyen de la combinaison d'un FFNN et d'un GNN qui permet de représenter, par des connexions résiduelles, les relations entre les niveaux GO non adjacents et les annotations des termes GO avec les gènes à tous les niveaux de l'ontologie. Ensuite, tous les niveaux de spécificité de l'ontologie sont représentés grâce à une loi de propagation inspirée des GNNs s'accompagnant d'une réduction du nombre de paramètres et facilitant leur intégration. À notre connaissance, seule la méthode ParsVNN intègre des connexions résiduelles dans un réseau FFNN. Néanmoins, dans cette méthode, une faible fraction de termes GO est intégrée, environ 28% du nombre total représenté dans GraphGONet. De plus, la part de gènes représentés dans les neurones de la couche d'entrée est de l'ordre du millier contre plus de 50 000 dans les données utilisées. Concernant les GNNs appliqués aux graphes de connaissances biologiques, les graphes IPP sont généralement exploités où seule l'entité représentant un gène est représentée et aucune autre entité biologique de sémantique supérieure telle qu'un terme GO. La conception originale de GraphGONet a permis de la rendre *self-explaining* alors que la plupart des méthodes d'intégration de connaissances ne le sont pas. Certes, ParsVNN a recours à une technique d'élagage pour proposer un graphe réduit par pathologie qui est un premier pas vers une interprétation globale du modèle, mais non personnalisable par individu. La couche de sélection proposée dans GraphGONet peut être considérée comme une technique d'élagage, sauf que contrairement à la précédente, elle permet d'avoir une sélection individualisée par patient qui forme le support de l'explication. De là, il est possible de comparer les signatures biologiques et obtenir une interprétation globale du modèle comme illustré dans le chapitre 5.

Finalement, la dernière méthode développée, BioHAN, est une extension de GraphGONet où différents graphes de connaissances sont exploités dans un graphe hétérogène par un GNN employant un mécanisme d'attention. Elle se distingue de l'état de l'art discuté du fait de l'intégration de connaissances multiples qui deviennent difficilement intégrables dans un FFNN tel qu'un MLP. La méthode BDKANN [Sno+21] dispose certes de deux couches cachées de neurones représentant deux sources de connaissances différentes, mais cet enchaînement dans un MLP reste limité si plusieurs niveaux d'une même source de connaissance sont intégrés. L'enchaînement de couches devient en effet moins propice, seul le recours à des techniques s'inspirant de la fusion multimodale [RT17] le permettrait. Nous pourrions ainsi concevoir différentes branches au sein du réseau de neurones à partir de la couche d'entrée où les couches cachées d'une branche représenteraient la structure d'une source de connaissance. Le signal partant de la couche d'entrée représentant les gènes serait dissipé dans chaque branche. Les signaux obtenus seraient ensuite concaténés dans une couche précédant la couche de sortie pour y réaliser la tâche de prédiction finale. Dans ce type d'architecture, il devient cependant moins évident de représenter les relations possibles entre ces différentes connaissances. L'architecture devient aussi très complexe et très gourmande en paramètres. De cette façon, les GNNs semblent plus appropriés, même si peu de méthodes ont été proposées sur des graphes hétérogènes où différentes entités et types de relations coexistent. Dans les deux premières approches, nous nous étions focalisées sur la notion de concept biologique représentant un niveau sémantique de haut niveau pour favoriser l'interprétation, ces concepts comme les fonctions biologiques peuvent avoir plus de sens pour un médecin que les gènes. De plus, les signatures basées sur les gènes peuvent s'avérer instables du fait que l'information y soit redondante et que plusieurs signatures soient possibles [DHZ12]. Au travers du travail mené sur GraphGONet, nous avons montré qu'au contraire, des mêmes fonctions biologiques ressortent même en apprenant plusieurs modèles. En passant ainsi par des concepts de plus haut niveau d'abstraction, le modèle réussit à apprendre des signatures consistantes. Néanmoins, il reste tout à fait possible d'identifier les gènes les plus importants qui pourraient intéresser des experts. Dans le cadre de Deep GONet, le calcul du score de pertinence des neu-

rones d'entrée peut se faire facilement à partir du score de pertinence des neurones de la première couche cachée. Dans GraphGONet, nous pourrions aussi avoir recours à une méthode a posteriori comme LRP pour remonter jusqu'à la couche de gènes. Au contraire, dans BioHAN, puisque nous avons recentralisé l'information autour du concept de gène en exploitant toujours les graphes de connaissances de plus haut niveau sémantique comme GO, nous pouvons mettre en évidence directement les gènes de plus grande importance et les concepts biologiques externes mobilisées. Dans tous les cas, les interprétations produites par l'ensemble de nos méthodes peuvent être facilement adaptées au profil de l'utilisateur. Enfin, BioHAN reste *self-explaining* et le recours à l'attention a l'avantage de favoriser l'interprétation des interactions dans le graphe et l'établissement de sous-graphes d'interprétation. Les approches *self-explaining* constituent un pas vers la conception de méthodes autosuffisantes ou exhaustives (*completeness* en anglais). Ce critère est décrit dans la littérature comme le fait de décrire le fonctionnement d'un système de manière précise, par exemple en exposant toutes les opérations mathématiques et tous les paramètres du système [Gil+18]. Le défi à relever est de faire en sorte que les explications restent accessibles et complètes. En ce qui concerne les méthodes à base de concepts, nous cherchons ainsi à déterminer si les ensembles de concepts choisis suffisent à expliquer les prédictions [Yeh+20]. Le modèle pourrait réellement être autosuffisant s'il a la capacité de s'affranchir d'une part totalement des tests statistiques d'enrichissement en connaissances externes et d'autre part des méthodes a posteriori relativement complexes et coûteuses. De plus, l'interprétation produite par ces méthodes a posteriori peut en effet, dans certains cas, ne pas s'aligner sur le raisonnement interne au modèle. Certaines méthodes a posteriori simples, peu paramétrables, peuvent toujours être utiles comme la méthode *Gradients*×*Inputs* [SVZ14]. BioHAN représente ainsi un pas vers cette autosuffisance.

7.2 Perspectives

Cette thèse ouvre sur de nombreuses perspectives autant sur la connaissance exploitée, les méthodes développées que sur les explications proposées. Quelques points ont déjà été abordés précédemment. Nous allons revenir sur certains et approfondir sur d'autres.

Tout d'abord, une différence principale entre Deep GONet et les deux autres méthodes concerne l'inclusion des gènes qui ne sont pas caractérisés dans les bases de connaissances utilisées (IPP, Reactome, GO). Cette intégration a été permise dans Deep GONet par les connexions noGO. Des solutions ont été proposées pour GraphGONet et BioHAN en s'appuyant notamment sur les graphes de coexpression, toutefois elles restent à être examinées. Dans les deux cas, si nous souhaitons maintenir le caractère *self-explaining* des modèles, l'intégration suppose de relier ces gènes aux termes d'annotations existants ou possiblement aux gènes déjà présents pour le cas de BioHAN. Si nous passons par un terme d'annotation virtuel pour relier ces gènes additionnels, l'interprétation sera moins facilitée. Notons néanmoins que BioHAN couvre plus de gènes que GraphGONet car les gènes non annotés par GO ou Reactome, mais présents dans le graphe IPP, ont été inclus. BioHAN reste comme une évolution de GraphGONet et Deep GONet pour répondre aux limites de ces dernières dont le manque d'expressivité des architectures proposées pour inclure diverses connaissances.

Ensuite, d'un point de vue sémantique, nous pourrions également prendre en compte les annotations qui peuvent exister, notamment sur les liaisons dans GO entre les termes ou entre un terme et un gène. GO est de plus en plus enrichi et de nouveaux réseaux d'annotations, intitulés GO-CAM (*GO-Causal Activity Model*) [1], offrent une vision plus globale des différentes interactions, habituellement séparées, entre les gènes et les termes issus des différentes sous-ontologies (MF, CC, BP) et même Reactome. Par exemple, un modèle GO-CAM peut représenter la façon dont les activités de différents produits génétiques fonctionnent ensemble dans une voie

1. Consultable à l'adresse : https://geneontology.cloud/home.

biologique. Nous pourrions nous servir de ce type de réseau pour enrichir les graphes utilisés dans BioHAN. La loi de propagation est facilement adaptable pour tenir compte de ces éléments sémantiques complémentaires.

De plus, nous avons essayé de proposer différents niveaux d'interprétation pour permettre d'évaluer la qualité d'interprétation de nos méthodes, qualifiée par certains de causabilité [HCM20]. Cette évaluation est essentielle pour d'une part, juger de la pertinence des explications et, d'autre part, vérifier que les prédictions du modèle se basent sur des caractéristiques cohérentes en accord avec la connaissance experte. Nous avons notamment montré que nous pouvions soumettre à validation les signatures biologiques retournées par nos modèles en examinant les liens possibles avec les phénotypes étudiés dans la littérature. Ces liens peuvent être extraits manuellement ou automatiquement par des outils TAL. Plus le nombre de liens est important, plus la prédiction est crédible. Quantifier ces liens pourrait servir à définir un seuil de fiabilité du modèle. Certains liens peuvent aussi participer à l'acquisition de nouvelles connaissances. Les scores de pertinence accompagnant ces signatures biologiques permettent également de mesurer leurs contributions finales à la prédiction. Certaines se démarquent plus que d'autres. Ces scores permettraient aussi d'évaluer la fiabilité du modèle en quantifiant l'incertitude du modèle, notamment lorsque le score indique que le signal est détourné de la sortie évaluée. La façon de mesurer la qualité d'interprétation d'un modèle n'est pas clairement définie dans la littérature. Toutefois, il existe quelques travaux à ce sujet. Doshi-Velez et Kim [DK18] ont par exemple proposé de formuler trois approches d'évaluation fondées respectivement sur l'application, l'humain et la fonctionnalité. Ces approches divergent selon la présence d'un sujet humain dans l'évaluation (*human in the loop* en anglais) et la difficulté de la tâche. La première, ancrée application, passe par une évaluation au travers d'expérimentations menées auprès des utilisateurs directs de l'application tels que des médecins dans le cas d'un outil de diagnostic médical. La seconde, fondée sur l'humain, vise à mener des expérimentations simples sur des sujets humains, pas forcément experts, en conservant l'objectif cible de l'application évaluée. La dernière, fonctionnellement centrée, n'a pas recours aux sujets humains et passent par des proxies qui peuvent prendre la forme de métriques adaptées au modèle expliqué pour l'évaluer. Un proxy régulièrement utilisé concerne la mesure de la parcimonie du modèle. En effet, plus un modèle est parcimonieux, moins il nécessite de paramètres, plus il est simple de l'interpréter. Généralement, nous mesurons dans le cas des systèmes à base de règles, le nombre de prémisses et dans le cas des arbres de décision, la profondeur. Le coût d'évaluation diminue de la première à la troisième approche. En effet, le recours à des testeurs (qualifiés ou non) peut s'avérer coûteux, entre autres pour mettre en place l'expérimentation. Dans le cas de la première approche, la vérification par un expert permettrait de vérifier la crédibilité des explications et si celles-ci répondent à leurs exigences [Ton+19 ; HCM20]. Néanmoins, en fonction du profil de l'utilisateur, les capacités à aborder l'explication et à se l'approprier peuvent être différentes [Nar+18]. Un modèle simple est donc parfois préféré. Comme nous avons essayé de le montrer dans les chapitres précédents, il est alors intéressant de proposer diverses formes d'explications en adéquation avec le profil de l'utilisateur. Nous pourrions les évaluer selon la seconde approche. Par ailleurs, certains travaux ont parfois directement inclus l'humain lors de l'optimisation du modèle pour en garantir son interprétation [Lag+18]. Néanmoins, une évaluation par l'homme reste coûteuse et subjective, car elle peut être biaisée par sa propre vision du monde. Un utilisateur peut aussi se laisser convaincre par une explication, corrompue, fournie en cas d'attaques adverses. Les modèles d'apprentissage profond sont en effet très sensibles aux attaques adverses souvent imperceptibles. Szegedy et al. [Sze+14] ont montré qu'une attaque adverse appliquée à une donnée d'image telle qu'un chien, initialement bien reconnue, est identifiée ensuite d'autruche. Au sujet de la troisième approche, nous avons cherché à atteindre l'objectif de parcimonie dans nos méthodes, soit en imposant une pénalité sur les poids des connexions, soit en réduisant le nombre de neurones et leurs concepts biologiques associés pour la prédiction finale. Nous travaillerons sur rendre plus parcimonieux les scores d'attention utilisés dans BioHAN. À propos des scores de pertinence, Ancona et al. [Anc+19] ont formulé

un ensemble de critères à évaluer :

- la *sensibilité-n*, telle que la somme des attributions pour tout sous-ensemble de caractéristiques de cardinalité n est égale à la variation de la sortie causée par la suppression des caractéristiques de ce sous-ensemble ;
- la *continuité*, telle que les entrées proches devraient avoir des scores de pertinence similaires ;
- l'*invariance* de l'implémentation, telle que différentes implémentations de la même méthode doivent conduire à des explications similaires des mêmes entrées.

Les méthodes d'attribution sont parfois utilisées en conjonction des méthodes de perturbation pour évaluer la stabilité et la sensibilité-n des signatures ressorties. Ces signatures sont perturbées de manière intelligente de façon à mesurer la corrélation entre les cartes de saillance et la sensibilité [Anc+18] ou la dégradation des performances en supprimant progressivement les signatures de plus grande pertinence [Sam+17]. On s'attend à une forte corrélation et une baisse significative des performances. Yeh et al. [Yeh+20], quant à eux, proposent une métrique d'évaluation basée sur SHAP, dénommée ConceptSHAP, pour mesurer la complétude, au niveau global ou en fonction d'une classe, d'un ensemble de concepts prédéfinis. Enfin, des critères ont été établis pour mesurer la qualité d'une explication dont la performance, la consistance, la compréhension et la certitude de prédiction [Mol20]. Nous avons montré que nos modèles respectent ces critères. L'ensemble de ces points ouvrent sur des pistes intéressantes pour évaluer de manière plus rigoureuse nos explications. Cependant, nous n'avons pas forcément accès à une vérité terrain. L'évaluation de l'interprétation d'un modèle représente ainsi un volet de recherche à part entière. Il existerait, enfin, en fonction des modèles d'apprentissage automatique utilisés, un dilemme entre interprétation et performance [Gun17]. En effet, on qualifie généralement les modèles à base d'apprentissage profond comme peu interprétables, mais performants, à l'opposé des arbres de décision qui sont relativement moins performants, mais plus interprétables. Un débat existe à ce sujet, où certains estiment que ce compromis ne devrait pas exister [Rud19]. Néanmoins, le fait qu'un modèle soit *self-explaining* et précis permettrait de lever ce dilemme. L'interprétation des réseaux de neurones est une propriété pour pouvoir bâtir une IA dite digne de confiance et responsable. Plusieurs autres propriétés [DK18 ; Gui+18], qui s'inscrivent dans le mouvement "Équité, responsabilité et transparence" (*Fair, accountable and transparent ML* (FATML) en anglais), sont à tenir en compte telles que la robustesse et la reproductibilité, la scabilité et la généralité à d'autres problèmes et d'autres données, l'usabilité [2] et l'utilité envers l'utilisateur, la causalité, la sobriété numérique [3], et l'adéquation à des standards éthiques (équité [4], vie privée). L'IA doit également s'inscrire dans le respect des objectifs de développement durable [5]. Un cadre législatif tend à paraître au niveau mondial pour éviter toutes dérives de l'utilisation de l'IA, de même que sur les données omiques où d'autres dérives telles que le transhumanisme ou l'eugénisme pourraient en découler. Les États membres de l'UNESCO ont, récemment, adopté en novembre 2021 le tout premier accord sur l'éthique de l'intelligence artificielle [6].

Finalement, un autre défi majeur de l'application des méthodes d'apprentissage profond sur les données d'expression de gènes concerne le manque de données. En effet, ces jeux de données

2. L'usabilité, ou l'aptitude à l'utilisation, se définit par la norme ISO 9241-11 comme « le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié. » Cette notion est distincte de l'utilité qui mesure à quel point les prédictions sont utiles aux utilisateurs et si ces derniers suivent ces prédictions.

3. Soucieuse de cet aspect, j'ai désiré faire le bilan carbone de mon doctorat présenté en Annexe 7.2 [Lac+19]

4. Les biais de discrimination peuvent exister comme l'avait montré Buolamwini [Buo17] dans une expérience sur un système d'identification faciale. Ce système devait déterminer si les visages étaient de sexe masculin ou féminin. Elle avait remarqué que sur une base de 100 visages, le logiciel avait eu du mal à identifier correctement les sujets féminins à peau foncée.

5. Un exemple d'initiative à ce sujet est le programme *AI for goods*.

6. https://fr.unesco.org/artificial-intelligence/ethics

contiennent très peu de patients (<5000), dû à leur coût élevé d'acquisition. Un exemple est la cohorte de données d'Oncodesign où le nombre de patients est inférieure à 250. À cause de ce faible nombre d'exemples, l'apprentissage des réseaux de neurones se heurte à des problèmes de surapprentissage. Dans l'article [HBZ22], nous avions remarqué qu'à partir d'un nombre suffisant d'exemples d'apprentissage, les modèles d'apprentissage profond surpassent les méthodes ML (SVM, RF...), mais en présence de peu de données, les performances sont moins bonnes. Cependant, nous avons identifié des scénarios où l'utilisation de jeux de données non liés au phénotype étudié provenant de larges bases de données publiques comme des données auxiliaires supervisées ou non supervisées peut permettre de pré-apprendre le réseau de neurones et l'aider à trouver une représentation adaptée des données d'expression de gènes. Par apprentissage par transfert, le réseau serait ajusté aux données cibles pour qui nous disposons de peu de patients, par exemple, atteints d'une maladie rare. Au cours de mon doctorat, j'ai eu l'opportunité d'encadrer des stages sur ce volet où l'objectif était d'adapter des méthodes de l'état de l'art issues des images telles que les méthodes d'apprentissage auto-supervisé [Goy+19]. Ces méthodes visent à pré-entraîner un modèle sur une tâche virtuelle. Il peut s'agir de rapprocher deux vues bruitées issues de la même image d'origine et d'éloigner celles qui proviennent d'images d'origine distinctes [Che+20b]. Des premiers résultats prometteurs en sont ressortis. Dans les travaux de cette thèse, nous n'avons pas réussi à montrer, peu importe la taille de la base d'apprentissage, une tendance générale où nos modèles seraient plus précis que les modèles opaques qui n'incluent aucune connaissance a priori. Il serait ainsi intéressant de mener des expériences d'apprentissage par transfert de nos modèles, interprétables par construction, pour évaluer l'apport de la connaissance a priori sur des tâches notamment plus complexes comme la survie.

Publications

[Bou+21a] Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune et Blaise Hanczar. "Deep GONet : self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data". In : *19th Asia Pacific Bioinformatics Conference (APBC 2021), National Cheng Kung University, Tainan, Taiwan, Feb 3-5, 2021*. 2021.

[Bou+21b] Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune et Blaise Hanczar. "Deep GONet : self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data". In : *Conférence sur l'Apprentissage automatique (CAp 2021), St Etienne, France, Jun 14-16, 2021*. 2021.

[Bou+21c] Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune et Blaise Hanczar. "Deep GONet : self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data". In : *BMC Bioinformatics* 22.10 (2021), p. 455. DOI : 10.1186/s12859-021-04370-7.

[BZH22a] Victoria Bourgeais, Farida Zehraoui et Blaise Hanczar. "GraphGONet : a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression". In : *Bioinformatics* (2022). DOI : 10.1093/bioinformatics/btac147.

[BZH22b] Victoria Bourgeais, Farida Zehraoui et Blaise Hanczar. "GraphGONet : a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression". In : *Journés Ouvertes Biologie, Informatique et Mathématiques (JOBIM 2022), Rennes, France, July 5-8, 2022*. 2022. URL : https://hal-univ-evry.archives-ouvertes.fr/hal-03608573.

[HBZ22] Blaise Hanczar, Victoria Bourgeais et Farida Zehraoui. "Assessment of deep learning and transfer learning for cancer prediction based on gene expression data". In : *BMC Bioinformatics* 23.1 (2022), p. 262. DOI : 10.1186/s12859-022-04807-7.

Bibliographie

Références pour le chapitre 1: Introduction

[19] "6 Non-coding RNA characterization". In: *Nature* (2019), pp. 1–1. DOI: 10.1038/nature28175 *cf. p. 2.*

[GBC16] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN: 978-0-262-03561-3. URL: http://www.deeplearningbook.org/ *cf. p. 3.*

[Gun17] David Gunning. "Explainable artificial intelligence (xai)". In: *Defense Advanced Research Projects Agency (DARPA), nd Web* 2.2 (2017), p. 1 *cf. p. 4, 105.*

[Kou+15] Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. "Machine learning applications in cancer prognosis and prediction". In: *Computational and structural biotechnology journal* 13 (2015), pp. 8–17. DOI: 10.1016/j.csbj.2014.11.005 *cf. p. 3.*

[NVR20] Bernard Nordlinger, Cédric Villani, and Daniela Rus, eds. *Healthcare and Artificial Intelligence*. Cham: Springer International Publishing, 2020. DOI: 10.1007/978-3-030-32161-1 *cf. p. 1.*

[SB75] Edward H. Shortliffe and Bruce G. Buchanan. "A model of inexact reasoning in medicine". In: *Mathematical Biosciences* 23.3 (1975), pp. 351–379. DOI: 10.1016/0025-5564(75)90047-4 *cf. p. 4.*

[Vil+18] Cédric Villani, Yann Bonnet, Charly Berthet, François Levin, Marc Schoenauer, Anne Charlotte Cornut, and Bertrand Rondepierre. *Donner un sens à l'intelligence artificielle: pour une stratégie nationale et européenne*. Conseil national du numérique, 2018 *cf. p. 1.*

[Zou+18] James Zou, Mikael Huss, Abubakar Abid, Pejman Mohammadi, Ali Torkamani, and Amalio Telenti. "A primer on deep learning in genomics". In: *Nature genetics* (2018), p. 1 *cf. p. 3.*

Références pour le chapitre 2: Préliminaires

[Bor+05] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. "Protein function prediction via graph kernels". In: *Bioinformatics* 21.Suppl_1 (2005), pp. i47–i56. DOI: 10.1093/bioinformatics/bti1007 *cf. p. 10.*

[Che+20a] Runpu Chen, Le Yang, Steve Goodison, and Yijun Sun. "Deep-learning approach to identifying cancer subtypes using high-dimensional genomic data". In: *Bioinformatics* 36.5 (2020), pp. 1476–1483. DOI: 10.1093/bioinformatics/btz769 *cf. p. 18.*

[DGH17] Padideh Danaee, Reza Ghaeini, and David Hendrix. "A Deep Learning Approach for Cancer Detection and Relevant Gene Identification". In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 22 (2017), pp. 219–229. DOI: 10.1142/9789813207813_0022 *cf. p. 17.*

[DM19] Maisa Daoud and Michael Mayo. "A survey of neural network-based cancer prediction models from microarray data". In: *Artificial Intelligence in Medicine* 97 (2019), pp. 204–214. DOI: 10.1016/j.artmed.2019.01.006 *cf. p. 17.*

[DGK07] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. "Weighted Graph Cuts without Eigenvectors A Multilevel Approach". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.11 (2007), pp. 1944–1957. DOI: 10.1109/TPAMI.2007.1115 *cf. p. 12.*

[Dwi+20] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. "Benchmarking Graph Neural Networks". 2020. arXiv: 2003.00982. URL: http://arxiv.org/abs/2003.00982 *cf. p. 11, 88, 91.*

[Fak+13] Rasool Fakoor, Faisal Ladhak, Azade Nazi, and Manfred Huber. "Using deep learning to enhance cancer diagnosis and classification". In: *Proceedings of the ICML Workshop on the Role of Machine Learning in Transforming Healthcare*. June 2013 *cf. p. 17.*

[GJ19] Hongyang Gao and Shuiwang Ji. "Graph U-Nets". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2083–2092. URL: http://proceedings.mlr.press/v97/gao19a.html *cf. p. 12.*

[Gil+17] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. "Neural message passing for Quantum chemistry". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. 2017, pp. 1263–1272. URL: http://proceedings.mlr.press/v70/gilmer17a.html *cf. p. 11.*

[Guy+17] "Inductive Representation Learning on Large Graphs". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 1024–1034. URL: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html *cf. p. 10, 11.*

[HBZ22] Blaise Hanczar, Victoria Bourgeais, and Farida Zehraoui. "Assessment of deep learning and transfer learning for cancer prediction based on gene expression data". In: *BMC Bioinformatics* 23.1 (2022), p. 262. DOI: 10.1186/s12859-022-04807-7 *cf. p. 18, 106.*

[Han+18] Blaise Hanczar, Mathieu Henriette, Toky Ratovomanana, and Farida Zehraoui. "Phenotypes Prediction from Gene Expression Data with Deep Multilayer Perceptron and Unsupervised Pre-training". In: *International Journal of Bioscience, Biochemistry and Bioinformatics* 8 (Jan. 2018), pp. 125–131 *cf. p. 18.*

[Han+20] Blaise Hanczar, Farida Zehraoui, Tina Issa, and Mathieu Arles. "Biological interpretation of deep neural network for phenotype prediction based on gene expression". In: *BMC Bioinformatics* 21.1 (2020). DOI: 10.1186/s12859-020-03836-4 *cf. p. 18, 33, 48, 58, 62, 66, 133.*

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 *cf. p. 8, 65, 88.*

[Hua+19] Jingjia Huang, Zhangheng Li, Nannan Li, Shan Liu, and Ge Li. "AttPool: Towards Hierarchical Feature Representation in Graph Convolutional Networks via Attention Mechanism". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, 2019, pp. 6479–6488. DOI: 10.1109/ICCV.2019.00658 *cf. p. 12, 92.*

[Kam+19] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. "Rethinking Knowledge Graph Propagation for Zero-Shot Learning". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11479–11488. DOI: 10.1109/CVPR.2019.01175 *cf. p. 11.*

[Kat+18] Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network". In: *BMC medical research methodology* 18.1 (2018), pp. 1–12. DOI: 10.1186/s12874-018-0482-1 *cf. p. 18.*

[KMR20] Tarek Khorshed, Mohamed N. Moustafa, and Ahmed Rafea. "Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions (GeneXNet)". In: *IEEE Access* 8 (2020), pp. 90615–90629. DOI: 10.1109/ACCESS.2020.2992907 *cf. p. 18.*

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015 *cf. p. 13.*

[KW14] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014 *cf. p. 9.*

[KW17] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl *cf. p. 11, 39, 81, 91.*

[Kol+15] Nikolay Kolesnikov et al. "ArrayExpress update—simplifying data submissions". In: *Nucleic Acids Research* 43.D1 (2015), pp. D1113–D1116. DOI: 10.1093/nar/gku1057 *cf. p. 15.*

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012 *cf. p. 18.*

[LLK19] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. "Self-Attention Graph Pooling". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 3734–3743. URL: http://proceedings.mlr.press/v97/lee19c.html *cf. p. 12, 92.*

[Li+17] Yuanyuan Li, Kai Kang, Juno M. Krahn, Nicole Croutwater, Kevin Lee, David M. Umbach, and Leping Li. "A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data". In: *BMC Genomics* 18 (2017). DOI: 10.1186/s12864-017-3906-0 *cf. p. 17.*

[McC+00] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. "Automating the Construction of Internet Portals with Machine Learning". In: *Information Retrieval* 3.2 (2000), pp. 127–163. DOI: 10.1023/A:1009953814988 *cf. p. 10.*

[Mos+20] Milad Mostavi, Yu-Chiao Chiu, Yufei Huang, and Yidong Chen. "Convolutional neural network models for cancer type prediction based on gene expression". In: *BMC medical genomics* 13.5 (2020), pp. 1–13 *cf. p. 18.*

[Mro+18] Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B. Tenenbaum, and Daniel L. K. Yamins. "Flexible Neural Representation for Physics Prediction". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 8813–8824 *cf. p. 12.*

[Ram+20] Ricardo Ramirez, Yu-Chiao Chiu, Allen Hererra, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. "Classification of Cancer Types Using Graph Convolutional Neural Networks". In: *Frontiers in Physics* 8.203 (2020), p. 203. DOI: 10.3389/fphy.2020.00203 *cf. p. 18, 39, 40, 82.*

[Rog+17] David S. Rogawski, Nicholas A. Vitanza, Angela C. Gauthier, Vijay Ramaswamy, and Carl Koschmann. "Integrating RNA sequencing into neuro-oncology practice". In: *Translational Research* 189 (2017), pp. 93–104. DOI: https://doi.org/10.1016/j.trsl.2017.06.013 *cf. p. 14.*

[TCF17] Vítor Teixeira, Rui Camacho, and Pedro G. Ferreira. "Learning influential genes on cancer gene expression data with stacked denoising autoencoders". In: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2017, pp. 1201–1205. DOI: 10.1109/BIBM.2017.8217828 *cf. p. 17.*

[TC21] Veronika Thost and Jie Chen. "Directed Acyclic Graph Neural Networks". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=JbuYF437WB6 *cf. p. 12, 81.*

[TH12] T Tieleman and G Hinton. "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning". In: *Tech. Rep., Technical report* (2012), p. 31 *cf. p. 13.*

[TCW15] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. "The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge". In: *Contemp Oncol* 19.1A (2015), pp. 68–77. DOI: 10.5114/wo.2014.47136 *cf. p. 16.*

[Tor+16] Aurora Torrente, Margus Lukk, Vincent Xue, Helen Parkinson, Johan Rung, and Alvis Brazma. "Identification of Cancer Related Genes Using a Comprehensive Map of Human Gene Expression". In: *PLOS ONE* 11.6 (2016), e0157484. DOI: 10.1371/journal.pone.0157484 *cf. p. 15.*

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008 *cf. p. 8, 22, 31, 88.*

[Vel+18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ *cf. p. 11, 87, 91.*

[Vin+08] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*. Ed. by William W. Cohen, Andrew McCallum, and Sam T. Roweis. Vol. 307. ACM International Conference Proceeding Series. ACM, 2008, pp. 1096–1103. DOI: 10.1145/1390156.1390294 *cf. p. 9.*

[WG18] Gregory P. Way and Casey S. Greene. "Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders". In: *Biocomputing 2018*. 2018, pp. 80–91. DOI: 10.1142/9789813235533_0008 *cf. p. 17.*

[Wu+20] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020), pp. 1–21. DOI: 10.1109/TNNLS.2020.2978386 *cf. p. 10.*

[Xin+21] Xiaohan Xing, Fan Yang, Hang Li, Jun Zhang, Yu Zhao, Mingxuan Gao, Junzhou Huang, and Jianhua Yao. "An Interpretable Multi-Level Enhanced Graph Attention Network for Disease Diagnosis with Gene Expression Data". In: *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2021, pp. 556–561. DOI: 10.1109/BIBM52615.2021.9669621 *cf. p. 18, 39, 40.*

[Xu+19] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. "How Powerful are Graph Neural Networks?" In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km *cf. p. 11.*

[Yin+18] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. "Hierarchical Graph Representation Learning with Differentiable Pooling". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 4800–4810 *cf. p. 12.*

[Zha+18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. "An End-to-End Deep Learning Architecture for Graph Classification". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 4438–4445. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17146 *cf. p. 12.*

[Zho+20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81. DOI: 10.1016/j.aiopen.2021.01.001 *cf. p. 10.*

Références pour le chapitre 3: État de l'art

[Ade+18] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. "Sanity Checks for Saliency Maps". In: *Advances in Neural Infor-*

mation Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 9525–9536. URL: https://proceedings.neurips.cc/paper/2018/hash/294a8ed24b1ad22ec2e7efea049b8737-Abstract.html *cf. p. 28.*

[Ahn+18] TaeJin Ahn, Taewan Goo, Chan-hee Lee, SungMin Kim, Kyullhee Han, Sangick Park, and Taesung Park. "Deep Learning-based Identification of Cancer or Normal Tissue using Gene Expression Data". In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Madrid, Spain: IEEE, 2018, pp. 1748–1752. DOI: 10.1109/BIBM.2018.8621108 *cf. p. 29.*

[AJ18] David Alvarez-Melis and Tommi S. Jaakkola. "Towards Robust Interpretability with Self-Explaining Neural Networks". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 7786–7795. URL: https://proceedings.neurips.cc/paper/2018/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html *cf. p. 22, 30.*

[Anc+19] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. "Gradient-Based Attribution Methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 169–191. DOI: 10.1007/978-3-030-28954-6_9 *cf. p. 27, 48, 104.*

[Bac+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7 (2015), e0130140. DOI: 10.1371/journal.pone.0130140 *cf. p. 22, 26, 46.*

[Bar+20] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012 *cf. p. 21.*

[Bau+17] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Network Dissection: Quantifying Interpretability of Deep Visual Representations". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, 2017, pp. 3319–3327. DOI: 10.1109/CVPR.2017.354 *cf. p. 22, 23.*

[Bea+] Valérie Beaudouin, Isabelle Bloch, David Bounie, Stéphan Clémençon, Florence d'Alché-Buc, James R Eagan, Winston Maxwell, Pavlo Mozharovskyi, and Jayneel Parekh. "Flexible and Context-Specific AI Explainability: A Multidisciplinary Approach". arXiv: 2003.07703. URL: https://arxiv.org/abs/2003.07703 *cf. p. 19, 20.*

[Bod+21] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. "Benchmarking and Survey of Explanation Methods for Black Box Models". 2021. arXiv: 2102.13076 *cf. p. 21.*

[Car00] Rich Caruana. "Case-Based Explanation for Artificial Neural Nets". In: *Artificial Neural Networks in Medicine and Biology, Proceedings of the ANNIMAB-1 Conference, Göteborg, Sweden, 13-16 May 2000*. Ed. by Helge Malmgren, Magnus Borga, and Lars Niklasson. Perspectives in Neural Computing. Springer, 2000, pp. 303–308. DOI: 10.1007/978-1-4471-0513-8_46 *cf. p. 22, 25.*

[Che+19] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan Su. "This Looks Like That: Deep Learning for Interpretable Image Recognition". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 8928–8939 *cf. p. 30.*

[CBR20] Zhi Chen, Yijie Bei, and Cynthia Rudin. "Concept whitening for interpretable image recognition". In: *Nature Machine Intelligence* 2.12 (2020), pp. 772–782. DOI: 10.1038/s42256-020-00265-z *cf. p. 30.*

[Che+21] Hryhorii Chereda, Annalen Bleckmann, Kerstin Menck, Júlia Perera-Bel, Philip Stegmaier, Florian Auer, Frank Kramer, Andreas Leha, and Tim Beißbarth. "Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer". In: *Genome Medicine* 13.1 (2021), p. 42. DOI: 10.1186/s13073-021-00845-7 *cf. p. 39, 40, 82.*

[CH19] Miruna-Adriana Clinciu and Helen Hastie. "A Survey of Explainable AI Terminology". In: *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2019)*. Association for Computational Linguistics, 2019, pp. 8–13. DOI: 10.18653/v1/W19-8403 *cf. p. 20.*

[Cog+08] John P. Cogswell et al. "Identification of miRNA changes in Alzheimer's disease brain and CSF yields putative biomarkers and insights into disease pathways". In: *Journal of Alzheimer's disease* 14.1 (2008), pp. 27–41. DOI: 10.3233/jad-2008-14103 *cf. p. 38.*

[Con04] Gene Ontology Consortium. "The Gene Ontology (GO) database and informatics resource". In: *Nucleic Acids Research* 32.suppl_1 (2004), pp. D258–D261. DOI: 10.1093/nar/gkh036 *cf. p. 33.*

[DG17] Piotr Dabkowski and Yarin Gal. "Real Time Image Saliency for Black Box Classifiers". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 6967–6976 *cf. p. 27.*

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett. 2016, pp. 3837–3845. URL: https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html *cf. p. 39.*

[DK18] Finale Doshi-Velez and Been Kim. "Considerations for Evaluation and Generalization in Interpretable Machine Learning". In: *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Ed. by Hugo Jair Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yagmur Güçlütürk, Umut Güçlü, and Marcel van Gerven. Cham: Springer International Publishing, 2018, pp. 3–17. DOI: 10.1007/978-3-319-98131-4_1 *cf. p. 19, 104, 105.*

[Elm21] Haitham Elmarakeby. "Data used in "Biologically informed deep neural network for prostate cancer discovery" Nature publication". Version 0.0.1. In: (2021). DOI: 10.5281/zenodo.5163213 *cf. p. 38.*

[Elm+21] Haitham A. Elmarakeby et al. "Biologically informed deep neural network for prostate cancer discovery". In: *Nature* 598.7880 (2021), pp. 348–352. DOI: 10.1038/s41586-021-03922-4 *cf. p. 35–38, 40.*

[Elt20] Daniel C. Elton. "Self-explaining AI as an Alternative to Interpretable AI". In: *Artificial General Intelligence - 13th International Conference, AGI 2020, St. Petersburg, Russia, September 16-19, 2020, Proceedings*. Ed. by Ben Goertzel, Aleksandr I. Panov, Alexey Potapov, and Roman Yampolskiy. Vol. 12177. Lecture Notes in Computer Science. Springer, 2020, pp. 95–106. DOI: 10.1007/978-3-030-52152-3_10 *cf. p. 29, 83.*

[Fab+18] Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, et al. "The reactome pathway knowledgebase". In: *Nucleic Acids Research* 46.D1 (2018), pp. D649–D655. DOI: 10.1093/nar/gkz1031 *cf. p. 33.*

[FFB19] Jelena Fiosina, Maksims Fiosins, and Stefan Bonn. "Explainable Deep Learning for Augmentation of Small RNA Expression Profiles". In: *Journal of Computational Biology* 27.2 (2019), pp. 234–247. DOI: 10.1089/cmb.2019.0320 *cf. p. 29.*

[FPV19] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. "Understanding Deep Networks via Extremal Perturbations and Smooth Masks". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South): IEEE, 2019, pp. 2950–2958. DOI: 10.1109/ICCV.2019.00304 *cf. p. 27, 28.*

[Fu94] LiMin Fu. "Rule generation from neural networks". In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.8 (1994), pp. 1114–1124 *cf. p. 25.*

[Gau+20] Thomas Gaudelet, Noël Malod-Dognin, Jon Sánchez-Valle, Vera Pancaldi, Alfonso Valencia, and Nataša Pržulj. "Unveiling new disease, pathway, and gene associations via multi-scale neural network". en. In: *PLOS ONE* 15.4 (2020), e0231059. DOI: 10.1371/journal.pone.0231059 *cf. p. 35, 36, 38, 40.*

[Gil+18] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. "Explaining Explanations: An Overview of Interpretability of Machine Learning". In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). 2018, pp. 80–89. DOI: 10.1109/DSAA.2018.00018 *cf. p. 19, 25, 103.*

[Hai16] Tameru Hailesilassie. "Rule Extraction Algorithm for Deep Neural Networks: A Review". In: *International Journal of Computer Science and Information Security* 14.7 (2016) *cf. p. 25.*

[Han+20] Blaise Hanczar, Farida Zehraoui, Tina Issa, and Mathieu Arles. "Biological interpretation of deep neural network for phenotype prediction based on gene expression". In: *BMC Bioinformatics* 21.1 (2020). DOI: 10.1186/s12859-020-03836-4 *cf. p. 18, 33, 48, 58, 62, 66, 133.*

[Hao+18] Jie Hao, Youngsoon Kim, Tae-Kyung Kim, and Mingon Kang. "PASNet: pathway-associated sparse deep neural network for prognosis prediction from high-throughput data". In: *BMC Bioinformatics* 19.1 (2018), p. 510. DOI: 10.1186/s12859-018-2500-z *cf. p. 35, 36, 38.*

[Hua+21] Xiaoqing Huang, Kun Huang, Travis Johnson, Milan Radovich, Jie Zhang, Jianzhu Ma, and Yijie Wang. "ParsVNN: parsimony visible neural networks for uncovering cancer-specific and drug-sensitive genes and pathways". In: *NAR Genomics and Bioinformatics* 3.4 (2021). DOI: 10.1093/nargab/lqab097 *cf. p. 35, 36, 38, 65, 66, 101.*

[JW19] Sarthak Jain and Byron C. Wallace. "Attention is not Explanation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 3543–3556. DOI: 10.18653/v1/N19-1357 *cf. p. 31, 32.*

[Jey+20] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. "How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 4211–4222 *cf. p. 22, 25.*

[Jin+21] Shuting Jin, Xiangxiang Zeng, Feng Xia, Wei Huang, and Xiangrong Liu. "Application of deep learning methods in biological networks". In: *Briefings in Bioinformatics* 22.2 (2021), pp. 1902–1917. DOI: 10.1093/bib/bbaa043 *cf. p. 39.*

[KG00] Minoru Kanehisa and Susumu Goto. "KEGG: kyoto encyclopedia of genes and genomes". In: *Nucleic Acids Research* 28.1 (2000), pp. 27–30. DOI: 10.1093/nar/28.1.27 *cf. p. 33.*

[Kan+17] Tianyu Kang, Wei Ding, Luoyan Zhang, Daniel Ziemek, and Kourosh Zarringhalam. "A biological network-based regularized artificial neural network model for robust phenotype prediction from gene expression data". In: *BMC Bioinformatics* 18.1 (2017), p. 565. DOI: 10.1186/s12859-017-1984-2 *cf. p. 35, 37, 38.*

[Kim+18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. PMLR, 2018, pp. 2673–2682. URL: http://proceedings.mlr.press/v80/kim18d.html *cf. p. 22–24.*

[Kin+19] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. "The (Un)reliability of Saliency Methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Vol. 11700. Lecture Notes in Computer Science. Springer, 2019, pp. 267–280. DOI: 10.1007/978-3-030-28954-6_14 *cf. p. 28.*

[Kin+18] Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. "Learning how to explain neural networks: PatternNet and PatternAttribution". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=Hkn7CBaTW *cf. p. 27.*

[KW17] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl *cf. p. 11, 39, 81, 91.*

[Koh+20a] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. "Concept Bottleneck Models". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5338–5348. URL: http://proceedings.mlr.press/v119/koh20a.html *cf. p. 22, 30, 31, 82, 101.*

[Koh+20b] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. "Towards Best Practice in Explaining Neural Network Decisions with LRP". In: (2020), pp. 1–7. DOI: 10.1109/IJCNN48605.2020.9206975 *cf. p. 27, 47.*

[Kue+20] Brent M. Kuenzi, Jisoo Park, Samson H. Fong, Kyle S. Sanchez, John Lee, Jason F. Kreisberg, Jianzhu Ma, and Trey Ideker. "Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells". In: *Cancer Cell* 38.5 (2020), 672–684.e6. DOI: 10.1016/j.ccell.2020.09.014 *cf. p. 36.*

[Li+18] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. "Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 3530–3537. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17082 *cf. p. 22, 30.*

[LL21] Chih-Hsu Lin and Olivier Lichtarge. "Using interpretable deep learning to model cancer dependencies". In: *Bioinformatics* btab137 (2021). DOI: 10.1093/bioinformatics/btab137 *cf. p. 35–38, 101.*

[Lip18] Zachary C. Lipton. "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57. DOI: 10.1145/3236386.3241340 *cf. p. 20, 21.*

[LL17] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf *cf. p. 22, 26.*

[LH18] Boyu Lyu and Anamul Haque. "Deep Learning Based Tumor Type Classification Using Gene Expression Data". In: *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. BCB '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 89–96. DOI: 10.1145/3233547.3233588 *cf. p. 33.*

[Mol20] Christoph Molnar. *Interpretable machine learning*. Lulu.com, 2020. URL: https://christophm.github.io/interpretable-ml-book *cf. p. 21, 105.*

[Mon+19] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. "Layer-Wise Relevance Propagation: An Overview". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Cham: Springer International Publishing, 2019, pp. 193–209. DOI: 10.1007/978-3-030-28954-6_10 *cf. p. 27, 47.*

[MSM18] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks". In: *Digital Signal Processing* 73 (2018), pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011 *cf. p. 19, 20, 26, 46–48.*

[ML22] Sehwan Moon and Hyunju Lee. "MOMA: a multi-task attention learning algorithm for multi-omics data interpretation and classification". In: *Bioinformatics* 38.8 (2022), pp. 2287–2296. DOI: 10.1093/bioinformatics/btac080 *cf. p. 32.*

[Ngu+16] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks". 2016. arXiv: 1605.09304. URL: http://arxiv.org/abs/1605.09304 *cf. p. 22, 23.*

[PMd21] Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché-Buc. "A Framework to Learn with Interpretation". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan. 2021, pp. 24273–24285 *cf. p. 22, 29, 30.*

[PWS19] Jiajie Peng, Xiaoyu Wang, and Xuequn Shang. "Combining gene ontology with deep neural networks to enhance the clustering of single cell RNA-Seq data". In: *BMC Bioinformatics* 20.8 (2019), p. 284. DOI: 10.1186/s12859-019-2769-6 *cf. p. 35, 36, 38, 64.*

[PDS18] Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 151 *cf. p. 27, 28.*

[Pol+14] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex". In: *Nature biotechnology* 32.10 (2014), pp. 1053–1058. DOI: doi.org/10.1038/nbt.2967 *cf. p. 38.*

[Qui+20] Thomas P. Quinn, Dang Nguyen, Santu Rana, Sunil Gupta, and Svetha Venkatesh. "DeepCoDA: personalized interpretability for compositional health data". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7877–7886 *cf. p. 32.*

[Ram+20] Ricardo Ramirez, Yu-Chiao Chiu, Allen Hererra, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. "Classification of Cancer Types Using Graph Convolutional Neural Networks". In: *Frontiers in Physics* 8.203 (2020), p. 203. DOI: 10.3389/fphy.2020.00203 *cf. p. 18, 39, 40, 82.*

[RSK18] SungMin Rhee, Seokjun Seo, and Sun Kim. "Hybrid Approach of Relation Network and Localized Graph Convolutional Filtering for Breast Cancer Subtype Classification". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 3527–3534. DOI: 10.24963/ijcai.2018/490 *cf. p. 39, 82.*

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778 *cf. p. 21, 22, 25.*

[Rud19] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215. DOI: 10.1038/s42256-019-0048-x *cf. p. 29, 105.*

[Sal94] Steven L Salzberg. *C4. 5: Programs for machine learning by J. Ross Quinlan. Morgan kaufmann publishers, inc., 1993*. 1994 *cf. p. 24.*

[SWM18] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Muller. "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models". In: *ITU Journal: ICT Discoveries* 1.1 (2018), pp. 39–48 *cf. p. 28.*

[Sel+17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74 *cf. p. 27.*

[Sha53] L. S. Shapley. "A Value for n-Person Games". In: *Contributions to the Theory of Games (AM-28), Volume II*. Ed. by Harold William Kuhn and Albert William Tucker. Princeton University Press, 1953, pp. 307–318. DOI: 10.1515/9781400881970-018 *cf. p. 25, 26.*

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. PMLR, 2017, pp. 3145–3153 *cf. p. 22, 26, 27.*

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014 *cf. p. 22, 23, 26, 47, 70, 103.*

[SGL20] Leon Sixt, Maximilian Granz, and Tim Landgraf. "When Explanations Lie: Why Many Modified BP Attributions Fail". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 9046–9057. URL: http://proceedings.mlr.press/v119/sixt20a.html *cf. p. 28.*

[Sla+20] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. "Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. AIES '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 180–186. DOI: 10.1145/3375627.3375830 *cf. p. 28.*

[Smi+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. "SmoothGrad: removing noise by adding noise". In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017 Workshop on Visualization for Deep Learning, Sydney, NSW, Australia, 6-11 August 2017. 2017. arXiv: 1706.03825. URL: http://arxiv.org/abs/1706.03825 *cf. p. 27.*

[Sne+00] Berend Snel, Gerrit Lehmann, Peer Bork, and Martijn A Huynen. "STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene". In: *Nucleic Acids Research* 28.18 (2000), pp. 3442–3444. DOI: 10.1093/nar/28.18.3442 *cf. p. 33.*

[Sno+21] Oliver Snow, Hossein Sharifi-Noghabi, Jialin Lu, Olga Zolotareva, Mark Lee, and Martin Ester. *Interpretable Drug Response Prediction using a Knowledge-based Neural Network*. Tech. rep. New York, NY, USA, 2021, pp. 3558–3568. DOI: 10.1145/3447548.3467212 *cf. p. 40, 102.*

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3319–3328 *cf. p. 22, 27, 47.*

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008 *cf. p. 8, 22, 31, 88.*

[VDH20] Sahil Verma, John Dickerson, and Keegan Hines. "Counterfactual Explanations for Machine Learning: A Review". 2020. arXiv: 2010.10596 *cf. p. 22, 28.*

[WL21] Tong Wang and Qihang Lin. "Hybrid Predictive Models: When an Interpretable Model Collaborates with a Black-box Model". In: *Journal of Machine Learning Research* 22 (2021), 137:1–137:38. URL: http://jmlr.org/papers/v22/19-325.html *cf. p. 22, 25.*

[Wan+20] Wei Wang, Xi Yang, Chengkun Wu, and Canqun Yang. "CGINet: graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph". In: *BMC Bioinformatics* 21.1 (2020), p. 544. DOI: 10.1186/s12859-020-03899-3 *cf. p. 40, 41, 87.*

[WP19] Sarah Wiegreffe and Yuval Pinter. "Attention is not not Explanation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. EMNLP-IJCNLP 2019. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 11–20. DOI: 10.18653/v1/D19-1002 *cf. p. 31.*

[Xin+21] Xiaohan Xing, Fan Yang, Hang Li, Jun Zhang, Yu Zhao, Mingxuan Gao, Junzhou Huang, and Jianhua Yao. "An Interpretable Multi-Level Enhanced Graph Attention Network for Disease Diagnosis with Gene Expression Data". In: *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2021, pp. 556–561. DOI: 10.1109/BIBM52615.2021.9669621 *cf. p. 18, 39, 40.*

[Yeh+20] Chih-Kuan Yeh, Been Kim, Sercan Ömer Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. "On Completeness-aware Concept-Based Explanations in Deep Neural Networks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020 *cf. p. 31, 103, 105.*

[YJS19] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. "INVASE: Instance-wise Variable Selection using Neural Networks". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=BJg%5C_roAcK7 *cf. p. 22, 29, 32.*

[Yu+18a] Michael K. Yu, Jianzhu Ma, Jasmin Fisher, Jason F. Kreisberg, Benjamin J. Raphael, and Trey Ideker. "Visible Machine Learning for Biomedicine". en. In: *Cell* 173.7 (2018), pp. 1562–1565. DOI: 10.1016/j.cell.2018.05.056 *cf. p. 34.*

[Yua+20] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. "Explainability in Graph Neural Networks: A Taxonomic Survey". In: *arXiv:2012.15445 [cs]* (2020). arXiv: 2012.15445. URL: https://arxiv.org/abs/2012.15445 *cf. p. 40, 82.*

[ZF14] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *Computer Vision – ECCV 2014*. Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Vol. 8689. Cham: Springer International Publishing, 2014, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53 *cf. p. 22, 24, 27.*

[Zha+21a] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. "Graph Neural Networks and Their Current Applications in Bioinformatics". In: *Frontiers in Genetics* 12 (2021), p. 690049 *cf. p. 39.*

[Zha+21b] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. "A Survey on Neural Network Interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021). Conference Name: IEEE Transactions on Emerging Topics in Computational Intelligence, pp. 726–742. DOI: 10.1109/TETCI.2021.3100641 *cf. p. 21, 25.*

[Zho+15] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. "Object Detectors Emerge in Deep Scene CNNs". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: https://people.csail.mit.edu/khosla/papers/iclr2015_zhou.pdf *cf. p. 22, 23, 27.*

[ZLJ16] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. "DeepRED – Rule Extraction from Deep Neural Networks". In: *Discovery Science*. Ed. by Toon Calders, Michelangelo Ceci, and Donato Malerba. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 457–473. DOI: 10.1007/978-3-319-46307-0_29 *cf. p. 22, 24.*

Références pour le chapitre 4: Deep GONet

[Alb+19] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. "iNNvestigate neural networks!" In: *Journal of Machine Learning Research* 20.93 (2019), pp. 1–8. URL: http://jmlr.org/papers/v20/18-540.html *cf. p. 54.*

[Anc+19] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. "Gradient-Based Attribution Methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 169–191. DOI: 10.1007/978-3-030-28954-6_9 *cf. p. 27, 48, 104.*

[Bac+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7 (2015), e0130140. DOI: 10.1371/journal.pone.0130140 *cf. p. 22, 26, 46.*

[Bhu+18] Khushwant S. Bhullar, Naiara Orrego Lagarón, Eileen M. McGowan, Indu Parmar, Amitabh Jha, Basil P. Hubbard, and H. P. Vasantha Rupasinghe. "Kinase-targeted cancer therapies: progress, challenges and future directions". In: *Molecular cancer* 17 (2018), p. 48. DOI: 10.1186/s12943-018-0804-2 *cf. p. 62.*

[Bou+21a] Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune, and Blaise Hanczar. "Deep GONet: self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data". In: *19th Asia Pacific Bioinformatics Conference (APBC 2021), National Cheng Kung University, Tainan, Taiwan, Feb 3-5, 2021*. 2021 *cf. p. 43.*

[Bou+21b] Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune, and Blaise Hanczar. "Deep GONet: self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data". In: *Conférence sur l'Apprentissage automatique (CAp 2021), St Etienne, France, Jun 14-16, 2021*. 2021 *cf. p. 43.*

[Bou+21c] Victoria Bourgeais, Farida Zehraoui, Mohamed Ben Hamdoune, and Blaise Hanczar. "Deep GONet: self-explainable deep neural network based on Gene Ontology for phenotype prediction from gene expression data". In: *BMC Bioinformatics* 22.10 (2021), p. 455. DOI: 10.1186/s12859-021-04370-7 *cf. p. 43.*

[CG14] S. Chockalingam and Siddhartha Sankar Ghosh. "Macrophage colony-stimulating factor and cancer: a review". In: *Tumor Biology* 35.11 (2014), pp. 10635–10644. DOI: 10.1007/s13277-014-2627-0 *cf. p. 60.*

[CC17] Francesco Ciccarese and Vincenzo Ciminale. "Escaping death: mitochondrial redox homeostasis in cancer cells". In: *Frontiers in oncology* 7 (2017), p. 117. DOI: 10.3389/fonc.2017.00117 *cf. p. 60.*

[Don19] Thomas Donoghue. "LISC: A Python Package for Scientific Literature Collection and Analysis". In: *Journal of Open Source Software* 4.41 (2019), p. 1674. DOI: 10.21105/joss.01674. URL: https://doi.org/10.21105/joss.01674 *cf. p. 66.*

[Gor04] J. Gorodkin. "Comparing two K-category assignments by a K-category correlation coefficient". In: *Computational Biology and Chemistry* 28.5 (2004), pp. 367–374. DOI: 10.1016/j.compbiolchem.2004.09.006 *cf. p. 49.*

[Han+20] Blaise Hanczar, Farida Zehraoui, Tina Issa, and Mathieu Arles. "Biological interpretation of deep neural network for phenotype prediction based on gene expression". In: *BMC Bioinformatics* 21.1 (2020). DOI: 10.1186/s12859-020-03836-4 *cf. p. 18, 33, 48, 58, 62, 66, 133.*

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 *cf. p. 8, 65, 88.*

[Hua+21] Xiaoqing Huang, Kun Huang, Travis Johnson, Milan Radovich, Jie Zhang, Jianzhu Ma, and Yijie Wang. "ParsVNN: parsimony visible neural networks for uncovering cancer-specific and drug-sensitive genes and pathways". In: *NAR Genomics and Bioinformatics* 3.4 (2021). DOI: 10.1093/nargab/lqab097 *cf. p. 35, 36, 38, 65, 66, 101.*

[JOV09] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. "Group Lasso with Overlap and Graph Lasso". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 433–440. DOI: 10.1145/1553374.1553431 *cf. p. 66.*

[Koh+20b] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. "Towards Best Practice in Explaining Neural Network Decisions with LRP". In: (2020), pp. 1–7. DOI: 10.1109/IJCNN48605.2020.9206975 *cf. p. 27, 47.*

[MR10] Vanina A Medina and Elena S Rivera. "Histamine receptors and cancer pharmacology". In: *British journal of pharmacology* 161.4 (2010), pp. 755–767. DOI: 10.1111/bph.14535 *cf. p. 60.*

[Mon+19] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. "Layer-Wise Relevance Propagation: An Overview". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Cham: Springer International Publishing, 2019, pp. 193–209. DOI: 10.1007/978-3-030-28954-6_10 *cf. p. 27, 47.*

[MSM18] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks". In: *Digital Signal Processing* 73 (2018), pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011 *cf. p. 19, 20, 26, 46–48.*

[PWS19] Jiajie Peng, Xiaoyu Wang, and Xuequn Shang. "Combining gene ontology with deep neural networks to enhance the clustering of single cell RNA-Seq data". In: *BMC Bioinformatics* 20.8 (2019), p. 284. DOI: 10.1186/s12859-019-2769-6 *cf. p. 35, 36, 38, 64.*

[RW17] Waseem Rawat and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9 (2017), pp. 2352–2449. DOI: 10.1162/neco_a_00990 *cf. p. 58.*

[Sam+19] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, eds. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. Cham: Springer International Publishing, 2019. DOI: 10.1007/978-3-030-28954-6 *cf. p. 47.*

[SB15] Richard Sever and Joan S. Brugge. "Signal Transduction in Cancer". In: *Cold Spring Harbor perspectives in medicine* 5.4 (2015), a006098. DOI: 10.1101/cshperspect.a006098 *cf. p. 60.*

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014 *cf. p. 22, 23, 26, 47, 70, 103.*

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3319–3328 *cf. p. 22, 27, 47.*

[Ton+20] Alexander Tong, David van Dijk, Jay S. Stanley III, Matthew Amodio, Kristina Yim, Rebecca Muhle, James Noonan, Guy Wolf, and Smita Krishnaswamy. "Interpretable Neuron Structuring with Graph Spectral Regularization". In: *Advances in Intelligent Data Analysis XVIII*. Ed. by Michael R. Berthold, Ad Feelders, and Georg Krempl. Cham: Springer International Publishing, 2020, pp. 509–521 *cf. p. 66.*

[YWC05] Hideki Yamaguchi, Jeffrey Wyckoff, and John Condeelis. "Cell migration in tumors". In: *Current opinion in cell biology* 17.5 (2005), pp. 559–564. DOI: 10.1016/j.ceb.2005.08.002 *cf. p. 62.*

[YB13] Ming Yang and William J. Brackenbury. "Membrane potential and cancer progression". In: *Frontiers in physiology* 4 (2013), p. 185. DOI: 10.3389/fphys.2013.00185 *cf. p. 60.*

[Yeo+21] Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. "Pruning by explaining: A novel criterion for deep neural network pruning". In: *Pattern Recognition* 115 (2021), p. 107899. DOI: 10.1016/j.patcog.2021.107899 *cf. p. 66.*

[Yu+18b] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. "Nisp: Pruning networks using neuron importance score propagation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9194–9203 *cf. p. 66.*

Références pour le chapitre 5: GraphGONet

[BZH22a] Victoria Bourgeais, Farida Zehraoui, and Blaise Hanczar. "GraphGONet: a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression". In: *Bioinformatics* (2022). DOI: 10.1093/bioinformatics/btac147 *cf. p. 67.*

[BZH22b] Victoria Bourgeais, Farida Zehraoui, and Blaise Hanczar. "GraphGONet: a self-explaining neural network encapsulating the Gene Ontology graph for phenotype prediction on gene expression". In: *Journés Ouvertes Biologie, Informatique et Mathématiques (JOBIM 2022), Rennes, France, July 5-8, 2022*. 2022. URL: https://hal-univ-evry.archives-ouvertes.fr/hal-03608573 *cf. p. 67.*

[Che+21] Hryhorii Chereda, Annalen Bleckmann, Kerstin Menck, Júlia Perera-Bel, Philip Stegmaier, Florian Auer, Frank Kramer, Andreas Leha, and Tim Beißbarth. "Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer". In: *Genome Medicine* 13.1 (2021), p. 42. DOI: 10.1186/s13073-021-00845-7 *cf. p. 39, 40, 82.*

[DHZ12] David Dernoncourt, Blaise Hanczar, and Jean-Daniel 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)*. 2012, pp. 350–355 *cf. p. 83, 102.*

[Elt20] Daniel C. Elton. "Self-explaining AI as an Alternative to Interpretable AI". In: *Artificial General Intelligence - 13th International Conference, AGI 2020, St. Petersburg, Russia, September 16-19, 2020, Proceedings*. Ed. by Ben Goertzel, Aleksandr I. Panov, Alexey Potapov, and Roman Yampolskiy. Vol. 12177. Lecture Notes in Computer Science. Springer, 2020, pp. 95–106. DOI: 10.1007/978-3-030-52152-3_10 *cf. p. 29, 83.*

[HCM20] Andreas Holzinger, André Carrington, and Heimo Müller. "Measuring the Quality of Explanations: The System Causability Scale (SCS)". In: *KI - Künstliche Intelligenz* 34.2 (2020), pp. 193–198. DOI: 10.1007/s13218-020-00636-z *cf. p. 84, 104.*

[KW17] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl *cf. p. 11, 39, 81, 91.*

[Koh+20a] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. "Concept Bottleneck Models". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5338–5348. URL: http://proceedings.mlr.press/v119/koh20a.html *cf. p. 22, 30, 31, 82, 101.*

[LL00] Scott W. Lowe and Athena W. Lin. "Apoptosis in cancer". In: *Carcinogenesis* 21.3 (2000), pp. 485–495. DOI: 10.3390/ijms19020448 *cf. p. 77.*

[Ram+20] Ricardo Ramirez, Yu-Chiao Chiu, Allen Hererra, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. "Classification of Cancer Types Using Graph Convolutional Neural Networks". In: *Frontiers in Physics* 8.203 (2020), p. 203. DOI: 10.3389/fphy.2020.00203 *cf. p. 18, 39, 40, 82.*

[RSK18] SungMin Rhee, Seokjun Seo, and Sun Kim. "Hybrid Approach of Relation Network and Localized Graph Convolutional Filtering for Breast Cancer Subtype Classification". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 3527–3534. DOI: 10.24963/ijcai.2018/490 *cf. p. 39, 82.*

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014 *cf. p. 22, 23, 26, 47, 70, 103.*

[TC21] Veronika Thost and Jie Chen. "Directed Acyclic Graph Neural Networks". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=JbuYF437WB6 *cf. p. 12, 81.*

[Yua+20] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. "Explainability in Graph Neural Networks: A Taxonomic Survey". In: *arXiv:2012.15445 [cs]* (2020). arXiv: 2012.15445. URL: https://arxiv.org/abs/2012.15445 *cf. p. 40, 82.*

Références pour le chapitre 6: BioHAN

[Dwi+20] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. "Benchmarking Graph Neural Networks". 2020. arXiv: 2003.00982. URL: http://arxiv.org/abs/2003.00982 *cf. p. 11, 88, 91.*

[GLJ21] H. Gao, Y. Liu, and S. Ji. "Topology-Aware Graph Pooling Networks". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 43.12 (2021), pp. 4512–4518. DOI: 10.1109/TPAMI.2021.3062794 *cf. p. 92.*

[Gra+21] Daniele Grattarola, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi. "Understanding Pooling in Graph Neural Networks". 2021. arXiv: 2110.05292. URL: http://arxiv.org/abs/2110.05292 *cf. p. 88, 92.*

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 *cf. p. 8, 65, 88.*

[Hua+19] Jingjia Huang, Zhangheng Li, Nannan Li, Shan Liu, and Ge Li. "AttPool: Towards Hierarchical Feature Representation in Graph Convolutional Networks via Attention Mechanism". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, 2019, pp. 6479–6488. DOI: 10.1109/ICCV.2019.00658 *cf. p. 12, 92.*

[KW17] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl *cf. p. 11, 39, 81, 91.*

[LLK19] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. "Self-Attention Graph Pooling". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 3734–3743. URL: http://proceedings.mlr.press/v97/lee19c.html *cf. p. 12, 92.*

[Mur+20] N. Murphy et al. "Insulin-like growth factor-1, insulin-like growth factor-binding protein-3, and breast cancer risk: observational and Mendelian randomization analyses with 430 000 women". In: *Annals of Oncology* 31.5 (2020). Publisher: Elsevier, pp. 641–649. DOI: 10.1016/j.annonc.2020.01.066 *cf. p. 97.*

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008 *cf. p. 8, 22, 31, 88.*

[Vel+18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ *cf. p. 11, 87, 91.*

[Wan+20] Wei Wang, Xi Yang, Chengkun Wu, and Canqun Yang. "CGINet: graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph". In: *BMC Bioinformatics* 21.1 (2020), p. 544. DOI: 10.1186/s12859-020-03899-3 *cf. p. 40, 41, 87.*

[YR00] Herbert Yu and Thomas Rohan. "Role of the Insulin-Like Growth Factor Family in Cancer Development and Progression". In: *JNCI: Journal of the National Cancer Institute* 92.18 (2000), pp. 1472–1489. DOI: 10.1093/jnci/92.18.1472 *cf. p. 97.*

Références pour le chapitre 7: Conclusion et perspectives

[Anc+18] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=Sy21R9JAW *cf. p. 105.*

[Anc+19] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. "Gradient-Based Attribution Methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 169–191. DOI: 10.1007/978-3-030-28954-6_9 *cf. p. 27, 48, 104.*

[Buo17] Joy Adowaa Buolamwini. "Gender shades : intersectional phenotypic and demographic evaluation of face datasets and gender classifiers". Massachusetts Institute of Technology, 2017. URL: https://dspace.mit.edu/handle/1721.1/114068 *cf. p. 105.*

[Che+20b] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. URL: http://proceedings.mlr.press/v119/chen20j.html *cf. p. 106.*

[DHZ12] David Dernoncourt, Blaise Hanczar, and Jean-Daniel 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)*. 2012, pp. 350–355 *cf. p. 83, 102.*

[DK18] Finale Doshi-Velez and Been Kim. "Considerations for Evaluation and Generalization in Interpretable Machine Learning". In: *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Ed. by Hugo Jair Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yagmur Güçlütürk, Umut Güçlü, and Marcel van Gerven. Cham: Springer International Publishing, 2018, pp. 3–17. DOI: 10.1007/978-3-319-98131-4_1 *cf. p. 19, 104, 105.*

[Gil+18] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. "Explaining Explanations: An Overview of Interpretability of Machine Learning". In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). 2018, pp. 80–89. DOI: 10.1109/DSAA.2018.00018 *cf. p. 19, 25, 103.*

[Goy+19] P. Goyal, D. Mahajan, A. Gupta, and I. Misra. "Scaling and Benchmarking Self-Supervised Visual Representation Learning". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019, pp. 6390–6399. DOI: 10.1109/ICCV.2019.00649 *cf. p. 106.*

[Gui+18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. "A Survey of Methods for Explaining Black Box Models". In: *ACM Computing Surveys* 51.5 (2018). DOI: 10.1145/3236009 *cf. p. 105.*

[Gun17] David Gunning. "Explainable artificial intelligence (xai)". In: *Defense Advanced Research Projects Agency (DARPA), nd Web* 2.2 (2017), p. 1 *cf. p. 4, 105.*

[HBZ22] Blaise Hanczar, Victoria Bourgeais, and Farida Zehraoui. "Assessment of deep learning and transfer learning for cancer prediction based on gene expression data". In: *BMC Bioinformatics* 23.1 (2022), p. 262. DOI: 10.1186/s12859-022-04807-7 *cf. p. 18, 106.*

[HCM20] Andreas Holzinger, André Carrington, and Heimo Müller. "Measuring the Quality of Explanations: The System Causability Scale (SCS)". In: *KI - Künstliche Intelligenz* 34.2 (2020), pp. 193–198. DOI: 10.1007/s13218-020-00636-z *cf. p. 84, 104.*

[Hua+21] Xiaoqing Huang, Kun Huang, Travis Johnson, Milan Radovich, Jie Zhang, Jianzhu Ma, and Yijie Wang. "ParsVNN: parsimony visible neural networks for uncovering cancer-specific and drug-sensitive genes and pathways". In: *NAR Genomics and Bioinformatics* 3.4 (2021). DOI: 10.1093/nargab/lqab097 *cf. p. 35, 36, 38, 65, 66, 101.*

[Koh+20a] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. "Concept Bottleneck Models". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5338–5348. URL: http://proceedings.mlr.press/v119/koh20a.html *cf. p. 22, 30, 31, 82, 101.*

[Lac+19] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. "Quantifying the Carbon Emissions of Machine Learning". In: *arXiv preprint arXiv:1910.09700* (2019) *cf. p. 105, 139.*

[Lag+18] Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. "Human-in-the-Loop Interpretability Prior". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018 *cf. p. 104.*

[LL21] Chih-Hsu Lin and Olivier Lichtarge. "Using interpretable deep learning to model cancer dependencies". In: *Bioinformatics* btab137 (2021). DOI: 10.1093/bioinformatics/btab137 *cf. p. 35–38, 101.*

[Mol20] Christoph Molnar. *Interpretable machine learning*. Lulu.com, 2020. URL: https://christophm.github.io/interpretable-ml-book *cf. p. 21, 105.*

[Nar+18] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. "How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation". 2018. arXiv: 1802.00682. URL: http://arxiv.org/abs/1802.00682 *cf. p. 104.*

[RT17] Dhanesh Ramachandram and Graham W. Taylor. "Deep Multimodal Learning: A Survey on Recent Advances and Trends". In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 96–108. DOI: 10.1109/MSP.2017.2738401 *cf. p. 102.*

[Rud19] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215. DOI: 10.1038/s42256-019-0048-x *cf. p. 29, 105.*

[Sam+17] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. "Evaluating the Visualization of What a Deep Neural Network Has Learned". In: *IEEE Trans. Neural Networks Learn. Syst.* 28.11 (2017), pp. 2660–2673. DOI: 10.1109/TNNLS.2016.2599820 *cf. p. 105.*

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014 *cf. p. 22, 23, 26, 47, 70, 103.*

[Sno+21] Oliver Snow, Hossein Sharifi-Noghabi, Jialin Lu, Olga Zolotareva, Mark Lee, and Martin Ester. *Interpretable Drug Response Prediction using a Knowledge-based Neural Network*. Tech. rep. New York, NY, USA, 2021, pp. 3558–3568. DOI: 10.1145/3447548.3467212 *cf. p. 40, 102.*

[Sze+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6199 *cf. p. 104.*

[Ton+19] Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. "What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use". In: *Proceedings of the 4th Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens. Vol. 106. Proceedings of Machine Learning Research. PMLR, 2019, pp. 359–380 *cf. p. 104.*

[Yeh+20] Chih-Kuan Yeh, Been Kim, Sercan Ömer Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. "On Completeness-aware Concept-Based Explanations in Deep Neural Networks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020 *cf. p. 31, 103, 105.*

Annexes

Annexe 1 : Données des cancers TCGA étudiés

BRCA Cancer du Sein (*Breast invasive carcinoma*)

HNSC Carcinome épidermoïde de la tête et du cou (*Head and Neck squamous cell carcinoma*)

KIRC Cancer du rein (*Kidney renal clear cell carcinoma*)

LGG Cancer du cerveau (*Brain lower grade glioma*)

LIHC Cancer du foie (*Liver hepatocellular carcinoma*)

LUAD Adénocarcinome pulmonaire (*Lung adenocarcinoma*)

LUSC Carcinome épidermoïde pulmonaire (*Lung squamous cell carcinoma*)

OV Cancer ovarien (*Ovarian serous cystadenocarcinoma*)

PRAD Adénocarcinome de la prostate (*Prostate adenocarcinoma*)

THCA Cancer de la thyroïde (*Thyroid carcinoma*)

UCEC Cancer du corps utérin (*Uterine corpus endometrial carcinoma*)

Annexe 2 : Deep GONet - résultats additionnels

Analyse de sensibilité

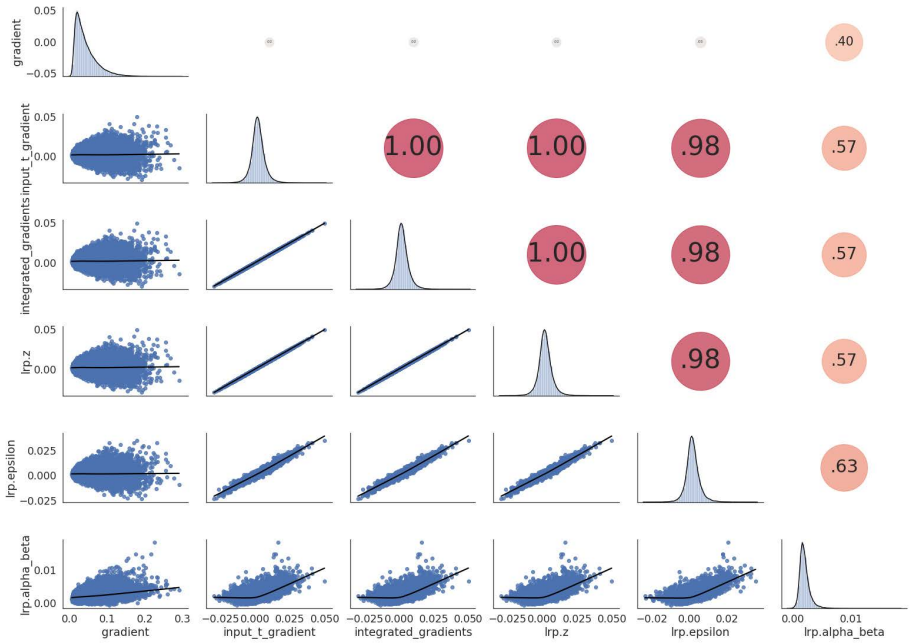


Figure 7.1 – ***Comparaison du score de pertinence des différentes méthodes d'attribution issue de l'annexe de [Han+20] (sous licence CC).*** *Pour chaque paire de méthodes, un graphique et la corrélation du score de pertinence sont donnés. La diagonale montre la distribution de la pertinence de chaque méthode.*

Interprétation biologique

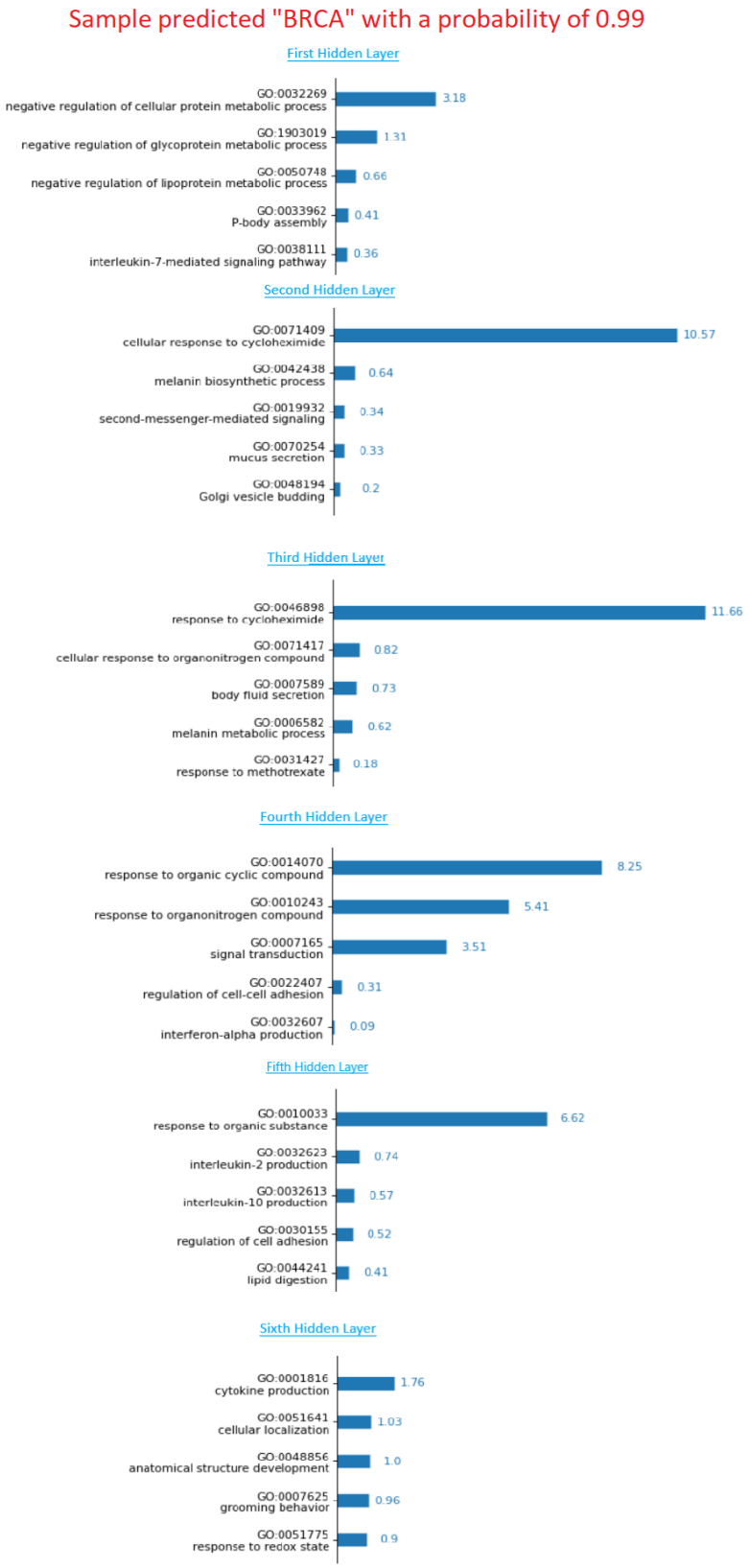
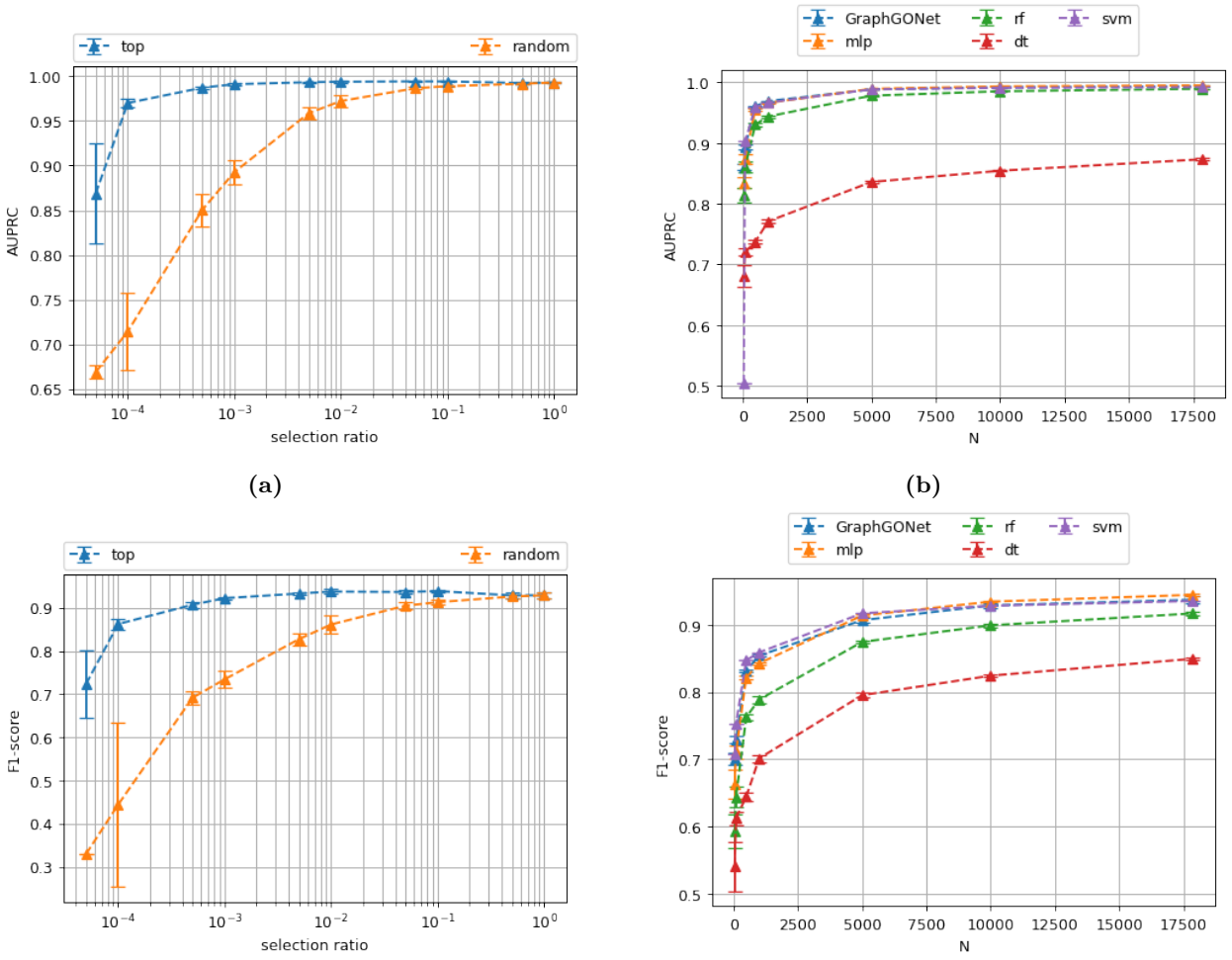


FIGURE 7.2 – ***Explication de la prédiction d'un patient BRCA.*** *À chaque couche, les termes GO sont classés en fonction de leur score de pertinence.*

Annexe 3 : GraphGONet - résultats additionnels

Analyse de sensibilité



(a) (b)

(c) (d)

FIGURE 7.3 – ***Évaluation de la performance des modèles (AUPCR et F1-Score) sur le jeu de données microarray selon (a-c) le ratio de sélection r et (b-d) le nombre d'exemples d'apprentissage N.***

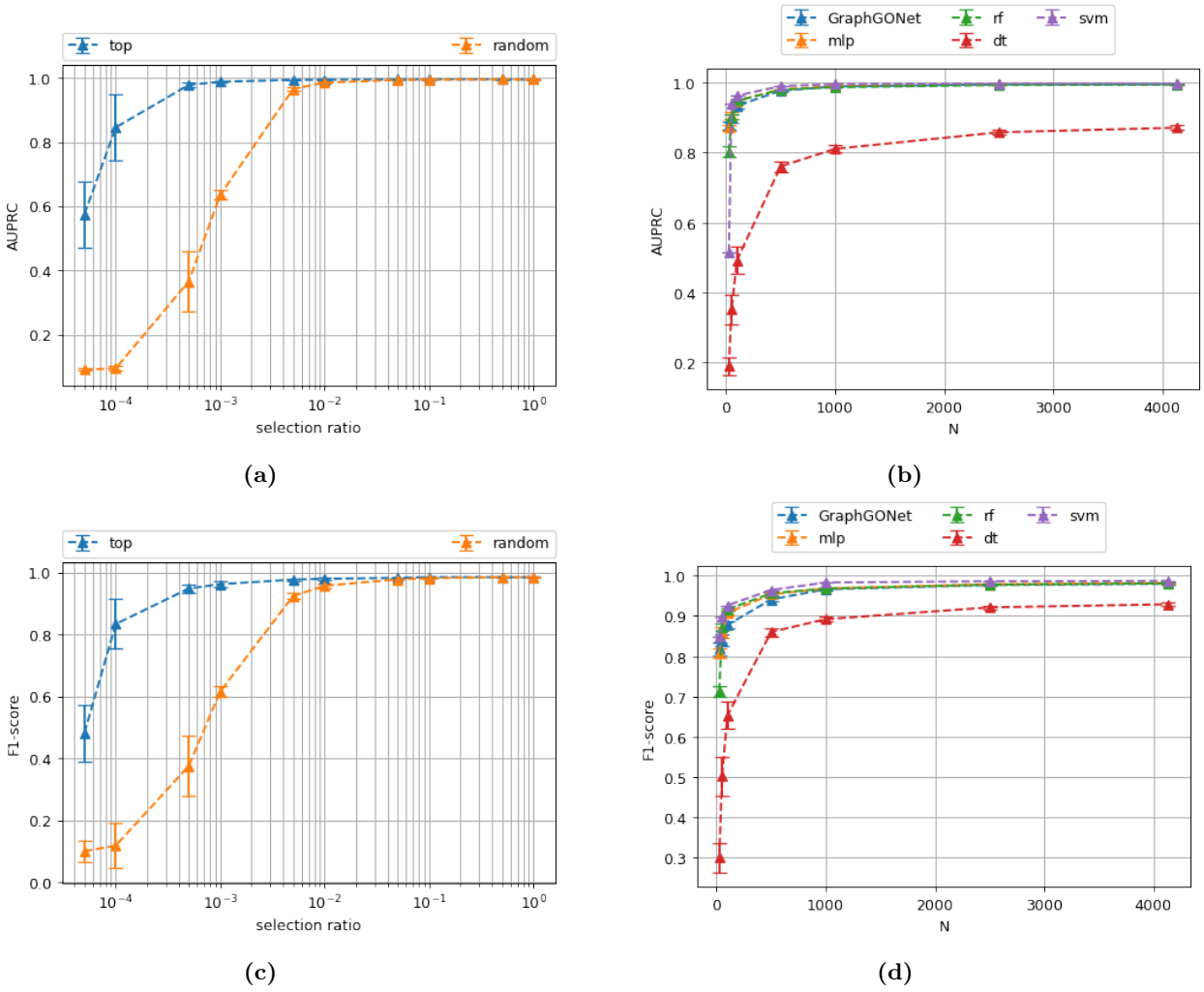
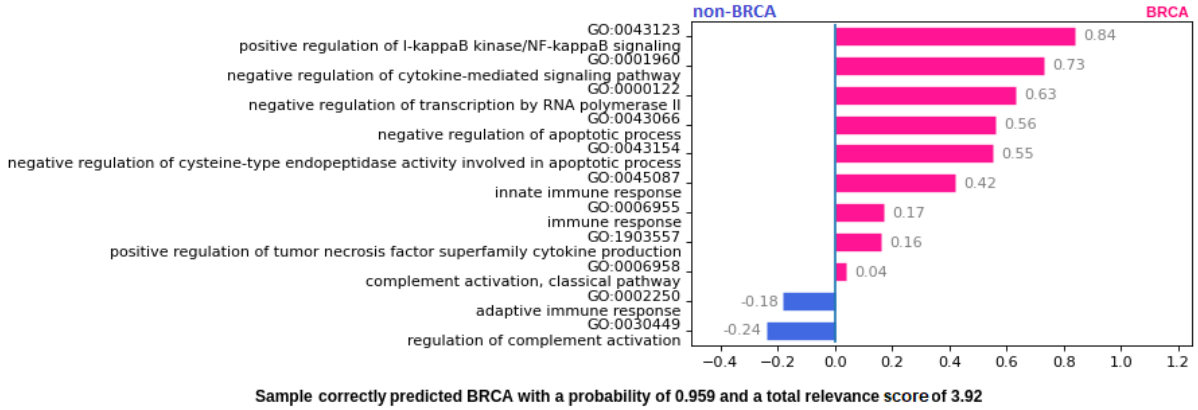


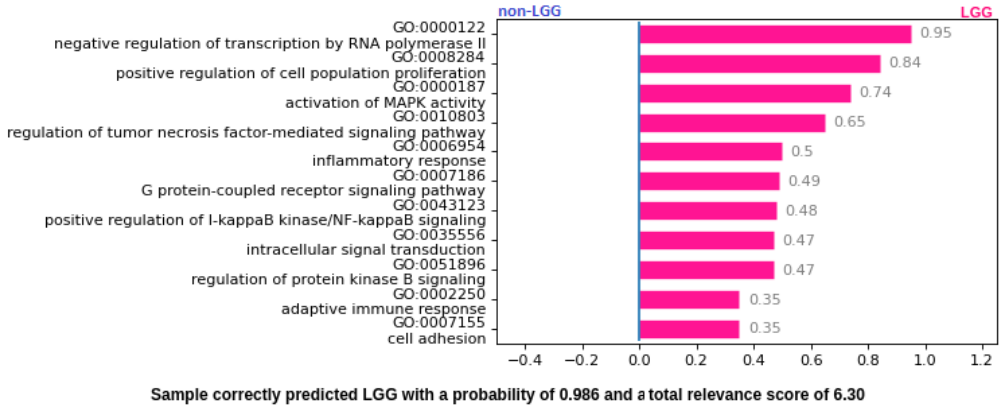
FIGURE 7.4 – ***Évaluation de la performance des modèles (AUPCR et F1-Score) sur le jeu de données TCGA selon (a-c) le ratio de sélection r et (b-d) le nombre d'exemples d'apprentissage N.***

Interprétation biologique

Les Figures 7.5(a-b) présentent une comparaison des explications fournies par GraphGONet sur deux patients du jeu de données TCGA de maladies différentes (respectivement BRCA et LGG). Dans les deux cas, les patients sont correctement prédits. Ces indicateurs montrent à nouveau que certains termes GO peuvent être importants pour différentes classes de résultats, mais avec des contributions quantitatives différentes. Par exemple, pour le terme "GO :0000122", le score de pertinence est de 0,63 pour le patient BRCA, contre 0,95 pour le patient LGG. En revanche, certains termes GO peuvent être spécifiques à certains tissus. Par exemple, le terme "GO :0007155", qui apparaît dans l'explication du patient LGG, n'est jamais sélectionné pour aucun des patients prédits BRCA. En outre, le terme "GO :0002250" n'est pas aussi significatif que les autres termes GO dans l'explication du patient LGG. Néanmoins, son impact est positif sur la prédiction finale, contrairement à l'explication du patient BRCA. Notez que la probabilité de prédiction du patient BRCA est plus faible que celle du patient LGG. Cela peut être expliqué par l'existence de scores de pertinence négatifs et par le fait que les contributions quantitatives des termes GO sont plus faibles dans le profil de pertinence du patient BRCA que dans celui du patient LGG.



(a) *Patient "BRCA"*



(b) *Patient "LGG"*

FIGURE 7.5 – ***Explication de (a) une prédiction BRCA et (b) une prédiction LGG.*** *Un ensemble de onze termes GO est présenté avec leur score de pertinence et leur description. La couleur indique vers quelle classe un terme GO influence le signal : bleu pour les autres classes et magenta pour la classe cible (LGG ou BRCA). Le score de pertinence total est la somme des scores de pertinence et du biais de la classe de sortie.*

Les figures suivantes présentent les dix termes GO les plus fréquents en fonction de leur occurrence sur les échantillons cancer/non-cancer et répondeur/non-répondeur. Nous pouvons observer que les termes GO les plus fréquents sont similaires, deux-à-deux, mais que la proportion de signes positifs-négatifs est différente. Par exemple, pour le terme GO "GO :0045944", le plus fréquent sur l'ensemble des profils, la proportion de signes positifs est supérieure de deux tiers (resp. trois quarts) à celle des signes négatifs pour le profil cancer (resp. répondeur), alors que c'est l'inverse pour le profil non-cancer (resp. non-répondeur). Même si ce n'est pas exactement le même classement, les chiffres sont complémentaires. Cela signifie qu'un terme GO peut être important pour les deux types de prédiction (même poids), mais que le signe du signal (activation) déterminera le résultat.

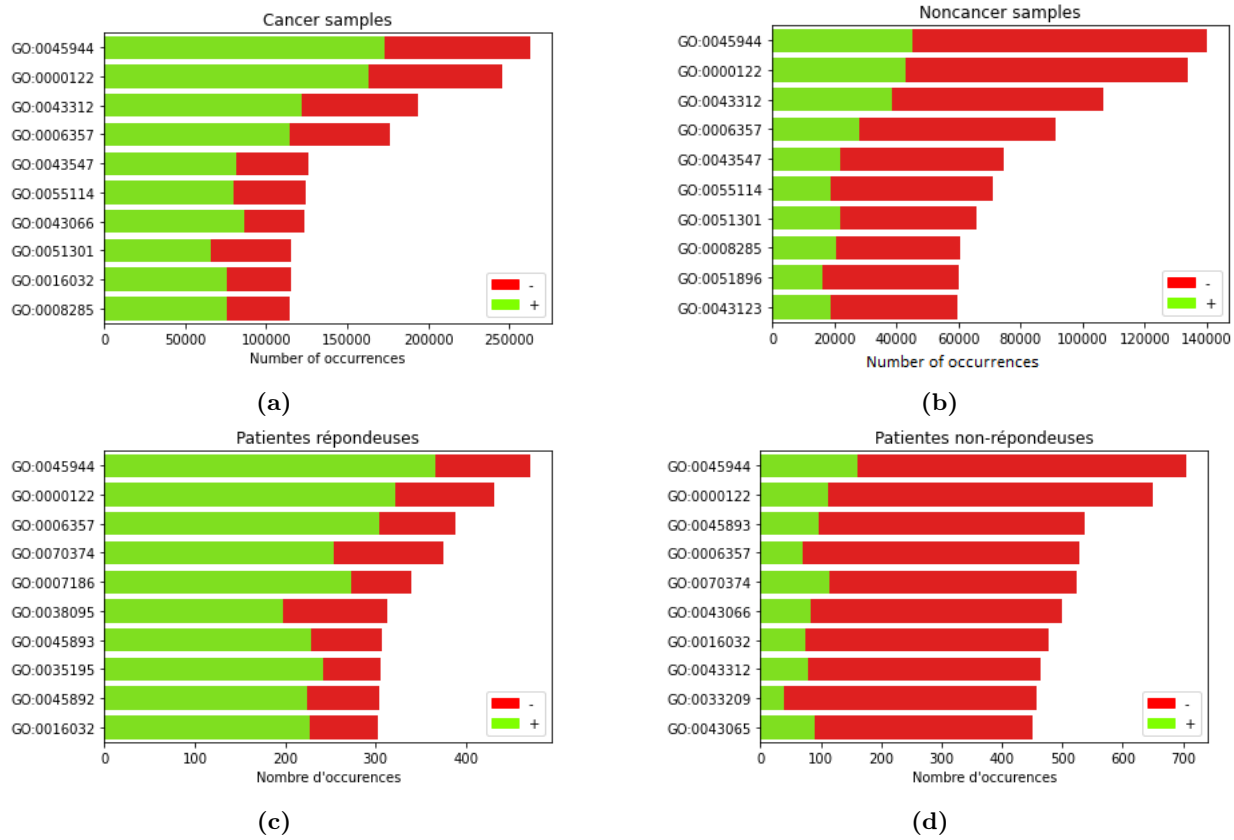


FIGURE 7.6 – ***Top-10 des termes GO les plus fréquents triés en fonction de leur occurrence pour (a-c) la prédiction cancer (resp. répondeur) et (b-d) non-cancer (resp. non-répondeur) sur l'ensemble de données microarray (resp. la cohorte d'OncoDesign).*** *Les couleurs indiquent la part des occurrences ayant un score de pertinence négatif (rouge) ou positif (vert). La fréquence maximale qui peut être atteinte correspond au nombre d'exemples cancer/non-cancer (resp. répondeur/non-répondeur) multiplié par le nombre de modèles, soit 368 800/189 000 (resp. 1300/1000).*

Annexe 4 : CO2 Emission Related to Experiments

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 $kgCO_2eq/kWh$. A cumulative of 10000 hours of computation was performed on hardware of type RTX 2080 Ti (TDP of 250W). Total emissions are estimated to be 1080 $kgCO_2eq$ of which 0 percents were directly offset.

Estimations were conducted using the Machine Learning Impact calculator presented in [Lac+19].

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Interprétation de l'apprentissage profond pour la prédiction de phénotypes à partir de données d'expression de gènes

Mots clés : données d'expression de gènes, apprentissage profond, médecine de précision, interprétation, connaissances a priori

Résumé : L'apprentissage profond est une avancée majeure de l'intelligence artificielle de ces dernières années. Ses domaines de prédilection sont principalement l'analyse d'image et le traitement du langage naturel. Un des futurs enjeux majeurs de cette approche est son application à la médecine de précision. Cette nouvelle forme de médecine permettra de personnaliser chaque étape du parcours de soin d'un patient en fonction de ses caractéristiques, notamment moléculaires telles que les données d'expression de gènes qui informent de l'état cellulaire d'un patient. Les modèles d'apprentissage profond sont néanmoins considérés comme des boîtes noires où aucune explication n'est fournie à la prédiction calculée. L'Union Européenne a adopté récemment un texte imposant aux algorithmes d'apprentissage automatique d'être capables d'expliquer leurs décisions aux utilisateurs. Il y a donc un réel besoin de rendre les réseaux de neurones plus interprétables et cela est particulièrement vrai dans le domaine médical pour différentes raisons. D'une part, pour s'assurer que le modèle se base sur des représentations fiables des patients et ne se concentre pas sur des artefacts non pertinents présents dans les données d'apprentissage. Ensuite, cela permettrait de rendre les différents utilisateurs (médecins, patients, chercheurs...) confiants dans leur utilisation de ce modèle. Enfin, un réseau de neurones performant pour la prédiction d'un certain phénotype peut avoir identifié une signature dans les données qui pourrait ouvrir sur de nouvelles pistes de recherche.

Dans l'état de l'art actuel, il existe deux approches pour interpréter les réseaux neurones : en créant des modèles qui sont par essence interprétables, ou en ayant recours a posteriori à une méthode tierce dédiée à l'interprétation du réseau de neurones déjà appris. Quelle que soit la méthode choisie, l'explication fournie consiste généralement en l'identification des variables d'entrée et des neurones importants pour la prédiction. Or, dans le cas d'une application sur les données d'expression de gènes, cela n'est pas suffisant, car ces données sont difficilement compréhensibles par l'homme. Nous proposons ainsi de nouvelles méthodes originales d'apprentissage profond, interprétables par construction. L'architecture de ces méthodes est définie à partir d'une ou plusieurs bases de connaissances. Un neurone y représente un objet biologique et les connexions entre les neurones correspondent aux relations entre les objets biologiques. Trois méthodes ont été développées, listées ci-dessous dans l'ordre chronologique.

La méthode Deep GONet se base sur un perceptron multicouche contraint par une base de connaissance biologique, la Gene Ontology (GO), par l'intermédiaire d'un terme de régularisation adapté. Les explications des prédictions sont fournies par une méthode d'interprétation a posteriori.

La méthode GraphGONet tire parti à la fois d'un perceptron multicouche et d'un réseau de neurones de graphes afin d'exploiter au maximum la richesse sémantique de la connaissance GO. Ce modèle a la capacité de rendre automatiquement des explications.

La méthode BioHAN ne se base plus que sur un réseau de neurones de graphes et peut facilement intégrer différentes bases de connaissances et leur sémantique. L'interprétation est facilitée par le recours aux mécanismes d'attention orientant le modèle à se concentrer sur les neurones les plus informatifs.

Ces méthodes ont été évaluées sur des tâches de diagnostic à partir de jeux de données d'expression de gènes réelles et ont montré leur compétitivité par rapport aux méthodes d'apprentissage automatique de l'état de l'art. Nos modèles fournissent des explications intelligibles composées des neurones les plus importants et des concepts biologiques qui leur sont associés. Cette caractéristique permet aux experts d'utiliser nos outils dans un cadre médical.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title : Interpretation of deep learning for phenotype prediction from gene expression data
Keywords : gene expression, deep learning, precision medicine, interpretation, prior knowledge

Abstract : Deep learning has been a significant advance in artificial intelligence in recent years. Its main domains of interest are image analysis and natural language processing. One of the major future challenges of this approach is its application to precision medicine. This new form of medicine will make it possible to personalize each stage of a patient's care pathway according to his or her characteristics, in particular molecular characteristics such as gene expression data that inform about the cellular state of a patient. However, deep learning models are considered black boxes as their predictions are not accompanied by an explanation, limiting their use in clinics. The General Data Protection Regulation (GDPR), adopted recently by the European Union, imposes that the machine learning algorithms must be able to explain their decisions to the users. Thus, there is a real need to make neural networks more interpretable, and this is particularly true in the medical field for several reasons. Understanding why a phenotype has been predicted is necessary to ensure that the prediction is based on reliable representations of the patients rather than on irrelevant artifacts present in the training data. Regardless of the model's effectiveness, this will affect any end user's decisions and confidence in the model. Finally, a neural network performing well for the prediction of a certain phenotype may have identified a signature in the data that could open up new research avenues.

In the current state of the art, two general approaches exist for interpreting these black-boxes : creating inherently interpretable models or using a third-party method dedicated to the interpretation of the trained neural network. Whatever approach is chosen, the explanation provided generally consists of identifying the important input variables and neurons for the prediction. However, in the context of phenotype prediction from gene expression, these approaches generally do not provide an understandable explanation, as these data are not directly comprehensible by humans. Therefore, we propose novel and original deep learning methods, interpretable by design. The architecture of these methods is defined from one or several knowledge databases. A neuron represents a biological object, and the connections between neurons correspond to the relations between biological objects. Three methods have been developed, listed below in chronological order.

Deep GONet is based on a multilayer perceptron constrained by a biological knowledge database, the Gene Ontology (GO), through an adapted regularization term. The explanations of the predictions are provided by a posteriori interpretation method.

GraphGONet takes advantage of both a multilayer perceptron and a graph neural network to deal with the semantic richness of GO knowledge. This model has the capacity to generate explanations automatically.

BioHAN is only established on a graph neural network and can easily integrate different knowledge databases and their semantics. Interpretation is facilitated by the use of an attention mechanism, enabling the model to focus on the most informative neurons.

These methods have been evaluated on diagnostic tasks using real gene expression datasets and have shown competitiveness with state-of-the-art machine learning methods. Our models provide intelligible explanations composed of the most contributive neurons and their associated biological concepts. This feature allows experts to use our tools in a medical setting.